Classification et inférence de réseaux pour les données RNA-seq

Mélina Gallopin

NNT : 2015SACLS174

Thèse de Doctorat
de
l'Université Paris-Saclay
préparée à
l'Université Paris-Sud 11

École doctorale n° 574
École Doctorale de Mathématiques Hadamard

Laboratoires d'accueil : Laboratoire de Mathématiques d'Orsay, UMR 8628 CNRS
Unité de Génétique Animale et Biologie Intégrative, UMR 1313 INRA

Spécialité : Mathématiques aux Interfaces

par

Mélina GALLOPIN

Classification et inférence de réseaux pour les données RNA-seq

présentée le 9 décembre 2015 au Département de Mathématiques d'Orsay, bât. 430, Salle Lederer.

Jury de soutenance :

Christophe AMBROISE	Professeur, Université Evry Val d'Essonne	Rapporteur
Gilles CELEUX	Directeur de recherche, Inria, Saclay	Directeur
Florence JAFFRÉZIC	Directrice de recherche, INRA, Jouy-en-Josas	Codirectrice
Marie-Laure MARTIN-MAGNIETTE	Directrice de recherche, IPS2, Paris-Saclay	Examinatrice
Pascal MASSART	Professeur, Université Paris-Sud	Président du jury
Grégory NUEL	Directeur de recherche, Université Pierre et Marie Curie	Examinateur
Andrea RAU	Chargée de Recherche, INRA, Jouy-en-Josas	Encadrante
Nathalie VILLA-VIALANEIX	Chargée de Recherche, INRA, Toulouse	Rapporteur

Thèse préparée au
Département de Mathématiques d'Orsay
Laboratoire de Mathématiques (UMR 8628), Bât. 425
Université Paris-Sud 11
91 405 Orsay CEDEX

Résumé

Cette thèse regroupe des contributions méthodologiques à l'analyse statistique des données issues des technologies de séquençage du transcriptome (RNA-seq). Les difficultés de modélisation des données de comptage RNA-seq sont liées à leur caractère discret et au faible nombre d'échantillons disponibles, limité par le coût financier du séquençage.

Une première partie de travaux de cette thèse porte sur la classification à l'aide de modèles de mélange. L'objectif de la classification est la détection de modules de gènes co-exprimés. Un choix naturel de modélisation des données RNA-seq est un modèle de mélange de lois de Poisson. Mais des transformations simples des données permettent de se ramener à un modèle de mélange de lois gaussiennes. Nous proposons de comparer, pour chaque jeu de données RNA-seq, les différentes modélisations à l'aide d'un critère objectif permettant de sélectionner la modélisation la plus adaptée aux données. Par ailleurs, nous présentons un critère de sélection de modèle prenant en compte des informations biologiques externes sur les gènes. Ce critère facilite l'obtention de classes biologiquement interprétables. Il n'est pas spécifique aux données RNA-seq. Il est utile à toute analyse de co-expression à l'aide de modèles de mélange visant à enrichir les bases de données d'annotations fonctionnelles des gènes.

Une seconde partie de travaux de cette thèse porte sur l'inférence de réseau à l'aide d'un modèle graphique. L'objectif de l'inférence de réseau est la détection des relations de dépendance entre les niveaux d'expression des gènes. Nous proposons un modèle d'inférence de réseau basé sur des lois de Poisson, prenant en compte le caractère discret et la grande variabilité inter-échantillons des données RNA-seq. Cependant, les méthodes d'inférence de réseau nécessitent un nombre d'échantillons élevé. Dans le cadre du modèle graphique gaussien, modèle concurrent au précédent, nous présentons une approche non-asymptotique pour sélectionner des sous-ensembles de gènes pertinents, en décomposant la matrice variance en blocs diagonaux. Cette méthode n'est pas spécifique aux données RNA-seq et permet de réduire la dimension de tout problème d'inférence de réseau basé sur le modèle graphique gaussien.

Mots-clefs : modèle de mélange, modèle graphique, données RNA-seq, classification, inférence de réseaux, sélection de modèle.

Clustering and network inference for RNA-seq data

Abstract

This thesis gathers methodologicals contributions to the statistical analysis of next-generation high-throughput transcriptome sequencing data (RNA-seq). RNA-seq data are discrete and the number of samples sequenced is usually small due to the cost of the technology. These two points are the main statistical challenges for modelling RNA-seq data.

The first part of the thesis is dedicated to the co-expression analysis of RNA-seq data using model-based clustering. A natural model for discrete RNA-seq data is a Poisson mixture model. However, a Gaussian mixture model in conjunction with a simple transformation applied to the data is a reasonable alternative. We propose to compare the two alternatives using a data-driven criterion to select the model that best fits each dataset. In addition, we present a model selection criterion to take into account external gene annotations. This model selection criterion is not specific to RNA-seq data. It is useful in any co-expression analysis using model-based clustering designed to enrich functional annotation databases.

The second part of the thesis is dedicated to network inference using graphical models. The aim of network inference is to detect relationships among genes based on their expression. We propose a network inference model based on a Poisson distribution taking into account the discrete nature and high inter sample variability of RNA-seq data. However, network inference methods require a large number of samples. For Gaussian graphical models, we propose a non-asymptotic approach to detect relevant subsets of genes based on a block-diagonale decomposition of the covariance matrix. This method is not specific to RNA-seq data and reduces the dimension of any network inference problem based on the Gaussian graphical model.

Keywords : mixture model, graphical model, RNA-seq data, clustering, network inference, model selection.

Remerciements

Je remercie Gilles Celeux d'avoir accepté de diriger cette thèse : merci Gilles pour vos conseils avisés et votre bienveillance. Je remercie Florence Jaffrézic d'avoir initié et dirigé cette thèse : merci Florence pour tes conseils et tes encouragements. Je remercie Andrea Rau d'avoir encadré cette thèse : Andrea, je te dois beaucoup, merci pour ton soutien, tes conseils et tes intuitions très utiles lorsqu'il s'agit de traiter des données réelles.

Je remercie également Emilie Devijver d'avoir accepté de collaborer dans l'écriture d'un chapitre de ma thèse : merci Emilie pour ton aide et pour nos discussions enrichissantes. Merci Gilles d'avoir initié ce travail collaboratif avec Emilie.

Je remercie Nathalie Villa-Vialaneix et Christophe Ambroise d'avoir accepté de rapporter ma thèse : merci Nathalie et Christophe pour votre lecture minutieuse et vos remarques pertinentes. Je remercie également Marie-Laure Martin-Magniette et Grégory Nuel d'avoir accepté d'examiner ma thèse, ainsi que Pascal Massart d'avoir accepté de présider mon jury de thèse.

Je remercie l'École Doctorale de Mathématiques de la région Paris Sud ED 142 et en particulier David Harrari d'avoir accepté de financer ma thèse appliquée. Je remercie Valérie Lavigne pour son efficacité et sa gentillesse lors de la dernière ligne droite avant la soutenance.

Je remercie les membres de l'Unité de Génétique Animale et Biologique Intégrative (GABI) de l'INRA de m'avoir accueillie dans leur locaux à Jouy-en-Josas, depuis mon stage de M2 jusqu'à ma soutenance. Je remercie en particulier Fabienne Le Provost, Sandrine Le Guillou, Yuliaxis Ramayo et Jordi Estelle pour avoir pris le temps de m'expliquer les aspects biologiques de ma recherche. Je remercie également tous les membres du réseau méthodologique MIA Inférence de réseaux, dont les réunions annuelles ont inspiré et rythmé ma thèse.

Je remercie Jean-Louis Fouley, Julien Chiquet, Laurent Schibler et Grégory Nuel pour leur conseils lors de mon comité de thèse. Je remercie également Jean-Baptiste Denis : merci pour ton aide précieuse qui m'a permis de surmonter une étape difficile de ma thèse.

Je remercie toutes les personnes rencontrées à l'occasion de mes trois missions doctorales "hors recherche" : d'abord au musée des sciences l'Exploradôme à Vitry-sur-Seine, puis lors de ma mission doctorale dédiée à la construction du MOOC de Bruno Falissard (merci en particulier à Bruno, Pauline et Marie-Gabrielle pour cette expérience semée d'embuches mais riche en enseignements), et enfin lors de mon monitorat au département GEA 2 de l'IUT de Sceaux.

Je remercie également les membres du MAP5 d'avoir accepté ma candidature en ATER à l'IUT Descartes, de me permettre de continuer à enseigner en IUT et de bénéficier de l'environnement de recherche du MAP5. Merci en particulier à Fabienne Comte, Jérôme Dedecker et Charles Bouveyron pour leur accueil. Merci également à Angelina Roche, Florence Muri, Servane Gey, Elisabeth Ottenwaelter, Rawya Zreik et Thomas Bastien pour leur aide à l'IUT.

Je remercie également toutes les personnes du bâtiments 211 de l'INRA qui ont

contribué à la bonne ambiance de ce lieu : Denis Laloë, Tatiana Zerjal, Dominique Montagu, Xavier Rognon, Agathe Vieaud, Michèle Boichard, Francis Minvielle, Patricia Huan, Wendy Brand-Williams, André Neau, Gwendal Restoux, Grégoire Leroy, Thomas Heams, Simon Boitard, Eléonore Charvolin, Yvelise Fricot, Marie-Hélène Pinard Vander-Laan, Valérie Bonjean, Nadjet Boushaba, Sonia Eynard, Fréderic Hospital et en particulier Gilles Monneret, Mathieu Tiret, Jean-Noël Hubert et Belén Jimenez pour leur aide et leur soutien, et François Deniau pour sa gentillesse.

Je remercie également les doctorants d'Orsay pour leur présence : Thomas Morzadec, Pierre-Antoine Guihéneuf, Vincent Pecastaing, Emilien Joly, Elodie Vernet, Pierre Gaillard, YueYuan Gao et Loïc Lacouture (merci pour ton accueil au MAP5). Merci en particulier à Solenne Thivin, Céline Abraham, Emilie Devijver et Valérie Robert pour leur aide et les bons moments passés en leur compagnie. Merci à ceux qui ont animé le couloir du 440 : Christine Keribin, Kevin Bleakley, Laura Brocco, Célia Barthélémy, Elodie Maillot, Romain Bar, Patrick Bouvier, Rémi Foucherau, Chú Cng Mp, Mohmoh A. Sedki, Sébastien Miquel, Florence Duc, Magda et Jana Khayal. Merci en particulier à Valérie Robert, Yann Vasseur, Vincent Brault et Jana Kalawoun pour l'entraide "fraternelle".

Enfin, je remercie ma famille pour son soutien et son amour inconditionnel ainsi que toutes les personnes qui m'ont aidée au cours de ces années de thèse et toutes les personnes qui comptent beaucoup pour moi : *con mucho cariño*, je pense à vous tendrement.

Note

Cette thèse est rédigée en français à l'exception des chapitres correspondant à des articles publiés ou en cours de publication.

— Les trois chapitres introductifs sont rédigés en français.

— Le chapitre 4 a fait l'objet d'une publication (Rau et al., 2013).

— Le chapitre 5, rédigé en français, a fait l'objet d'une communication orale aux $47^{ième}$ *Journées de la Statistique* à Lille en juin 2015.

— Le chapitre 6 a fait l'objet d'une publication acceptée (Gallopin et al., 2015).

— Le chapitre 7 a fait l'objet d'une publication (Gallopin et al., 2013).

— Le chapitre 8 fait l'objet d'un article en cours d'écriture en collaboration avec Emilie Devijver.

— La conclusion est rédigée en français.

Table des matières

Chapitre 1

Le contexte biologique

1.1 La technologie de séquençage haut-débit RNA-seq

1.1.1 Mesurer l'expression des gènes

L'acide désoxyribonucléique, ou ADN, est une macromolécule constituée d'un enchainement de quatre types d'acides aminés : adénine (A), cytosine (C), guanine (G) ou thymine (T). Cette molécule est contenue dans chaque cellule des êtres vivants. Elle contient l'information génétique, appelé le génome, pour chaque individu. Le génome désigne l'ensemble du matériel génétique d'un individu codé dans son ADN. Dans la cellule, l'ADN est structuré sous forme de chromosomes et certaines zones de cette molécule d'ADN, appelé des gènes, représentent des régions codantes qui traduisent en ARN puis protéines l'information contenue dans l'ADN. Ces zones sont appelées les gènes. Le dogme central de la biologie moléculaire schématisé en Figure 1.1 décrit le mécanisme d'expression des gènes. De manière schématique, une molécule d'acide ribonucléique (ARN) est transcrite à partir d'un gène, puis une protéine est traduite à partir de cet ARN. Le passage de la molécule d'ADN à la molécule d'ARN est la *transcription*, réalisé à l'aide de l'ARN polymérase. Le passage de la molécule d'ARN à la protéine est la *traduction*. Le transcriptome désigne l'ensemble des molécules d'ARN présentes dans la cellule.

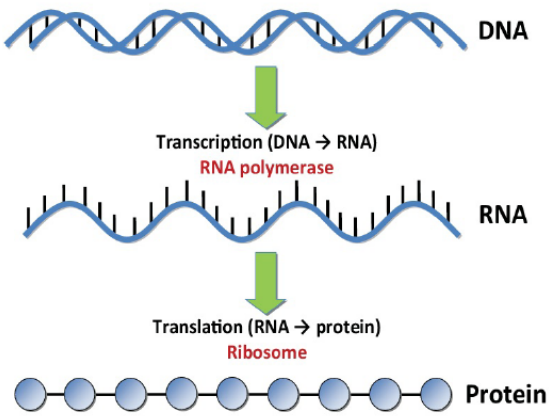

Figure 1.1 – Schéma du dogme central de la biologie moléculaire. Figure extraite de Rau (2010).

Le premier génome séquencé est celui d'une bactérie durant les années 1970 (Sanger et al., 1977). Ce génome ne comportait que 5 000 nucléotides et le coût de séquençage était très élevé. Les méthodes ont depuis beaucoup évolué, notamment grâce au développement de la technique de réaction en chaîne par polymérase (PCR pour *polymerase chain reaction*), qui permet de répliquer artificiellement un brin d'ADN. Cette technique de PCR est toujours très utilisée. On séquence à l'heure actuelle le génome de nombreux individus à un prix toujours décroissant. En septembre 2001, le coût de séquençage d'un génome complet humain s'élevait à presque 100 millions de dollars. En avril 2015, ce coût est d'environ 4000 dollars [1]. Cette baisse du coût de séquençage de l'ADN entraine

1. http://www.genome.gov/sequencingcosts/

la production de très grandes quantités de données : Stephens et al. (2015) prédisent que entre 100 millions et 2 milliards de génomes humains seront séquencés d'ici 2025. Cependant, connaître la séquence d'ADN d'un individu ne suffit pas pour comprendre les mécanismes d'expression de ses gènes. Pour les comprendre, le contenu en ARN et protéines des cellules doit être analysé.

Les protéines sont les acteurs principaux des phénomènes biologiques observés et leur présence ou absence peut expliquer une caractéristique précise comme un phénotype ou une pathologie. Cependant, mesurer la quantité des protéines est très difficile du fait de leur instabilité et de leur dégradation rapide dans la cellule. Il est de plus difficile d'associer une protéine donnée à une position du génome. La mesure de la quantité d'ARN matures, stables, est plus facile. Cette quantité de transcrits est donc considérée comme représentative du niveau d'expression du gène (*i.e.* de l'abondance des protéines produites). Dans cette thèse, l'expression des gènes désigne la quantité de transcrits présents dans la cellule associée à l'ensemble des gènes considérés.

Pour mesurer cette quantité de transcrits, plusieurs méthodes existent et suivent le protocole général suivant. En amont, les conditions expérimentales et le nombre d'échantillons par condition expérimentale doivent être déterminés en fonction du type d'analyses souhaitées. L'ARN est ensuite extrait de la cellule et retranscrit en un brin d'ADN complémentaire (ADNc) à l'aide d'enzymes spécifiques. Le brin complémentaire de l'ADNc est également produit, puis l'ARN présent est retiré du mélange. Seuls les ARN matures sont collectés. Par la suite, les brins d'ADNc sont amplifiés par PCR afin de disposer d'une quantité assez importante pour la suite du protocole. Il convient alors de mesurer la quantité d'ADNc. Pour effectuer cette mesure, plusieurs méthodes existent :

Les puces à ADN Cette technique de mesure, aussi appelée technologie *microarrays*, est apparue dans les années 1990. Les brins d'ADNc sont marqués par une teinture fluorescente ou radioactive. Parallèlement, une puce à ADN est préparée contenant les portions d'ADN de l'individu dont on souhaite mesurer l'expression. Les brins d'ADNc s'hybrident aux ADN présents sur la puce : les nucléotides des brins d'ADNc viennent s'unir aux nucléotides des brins d'ADN de la puce à ADN par complémentarité. Le niveau d'hybridation est alors détecté par radioactivité ou fluorescence pour les différentes conditions expérimentales. L'expression mesurée correspond donc à un niveau de fluorescence différentiellement mesuré entre deux conditions expérimentales pour les puces à deux couleurs, ou à l'abondance de chaque ADN ciblé pour les puces à une couleur. Cette mesure d'abondance est continue et relative. Le niveau de précision de la méthode dépend du nombre de fragments d'ADN attachés sur la puce à ADN. Elle nécessite de connaître par avance la séquence d'ADN que l'on souhaite hybrider. Cette technologie ne permet pas de découvrir de nouvelles régions codantes du génome.

La technologie SAGE (*Serial Analysis of Gene Expression*) : Cette technique a été mise au point dans les années 1990. Contrairement à la technologie des puces à ADN, la technologie SAGE ne requiert pas la connaissance du génome de l'individu pour être utilisée. Des séquences codantes d'ADNc (de 9 à 14 paires de bases nucléiques), appelées étiquettes ou *tags* en anglais, sont séquencées à partir des brins d'ADNc. Ces *tags* sont ensuite recollés bout à bout pour former une molécule d'ADN synthétique, et ainsi révéler la séquence d'ADN (et donc le gène) dont elle est issue. En effectuant cette opération un grand nombre de fois,

on peut ainsi mesurer l'expression des gènes. Cette technologie, plus coûteuse que la technologie des puces à ADN, est plus précise.

La technologie *RNA-sequencing* (RNA-seq) : Comme la technologie SAGE, la technologie de RNA-seq est basée sur le séquencage de l'ADNc et non une technique d'hybridation de l'ADNc comme la technologie des puces à ADN. Cette technique peut être utilisée pour séquencer directement l'ADN d'un individu (DNA-seq) afin de créer un génome de référence ou pour détecter des altérations de la séquence d'ADN. Le RNA-seq est une technique de séquençage de *seconde génération*.

Dans cette thèse, nous nous intéressons exclusivement aux données issues de la technologie RNA-seq.

1.1.2 Le RNA-seq en pratique

Nous décrivons les étapes du séquençage de l'ADNc par la technologie RNA-seq, en insistant sur les diverses formes de la collecte de données.

1. **Lecture des brins d'ADN :** L'ADNc est coupé en petit morceaux de longueur de 200 à 300 paires de bases. Cette longueur dépend du séquenceur utilisé (e.g. Illumina, 454) et évolue progressivement vers des reads de plus en plus longs. Les petits morceaux d'ADNc sont lus. Les lectures correspondantes sont appelées des *reads*. Les séquences codantes pour chaque *read* sont répertoriées dans un fichier texte au format `FASTQ`. Ce fichier contient également des informations sur la qualité du séquençage, comme décrit dans la Figure 1.2.

```
@IDREAD
AATGTCACGTCGCTGATCGATGCTTAGTTTTCGGCGCGCATTGCGCTGAAA
+
;;3;;;;;;;>>>;;;;;7;;;;;;;88
```

Figure 1.2 – Extrait du descriptif d'un *read* dans un fichier `.FASTQ`. La première ligne contient l'identifiant du *read* précédé du signe @, la seconde contient la séquence d'ADN, les lignes 3 et 4 permettent l'identification et le contrôle de qualité du *read*.

2. **Contrôle de qualité du séquençage :** Les fichiers `.FASTQ` sont nettoyés : les *reads* de mauvaise qualité sont supprimés ou rétrécits (*trimming*). Le programme `fastQC`[2] permet d'effectuer cette étape.
3. **Alignement des *reads* :** Si l'on dispose d'un génome de référence, les *reads* sont alignés sur ce génome de référence. Le génome de référence est idéalement la séquence complète d'ADN à partir de laquelle ont été produit les ARN. Plus généralement, il s'agit de la séquence d'ADN type de l'espèce étudiée disponible sous la forme de fichier `.fa` disponible en téléchargement en ligne depuis le site NCBI[3]. Si aucun génome de référence n'est disponible, les *reads* sont assemblés pour former des *contigs*. Aligner un *read* sur un génome de référence consiste à

2. http ://www.bioinformatics.babraham.ac.uk/projects/fastqc/
3. http ://hgdownload.soe.ucsc.edu

chercher la portion du génome de référence la plus ressemblante possible au *read*, comme illustré sur la Figure 1.3.

FIGURE 1.3 – Schéma d'alignement de deux *reads* (en rouge et bleu) sur un génome de référence. En réalité, les *reads* séquencés sont longs d'au moins 50 bases (A,C, T ou G). L'alignement est rarement parfait, un nombre de bases peut différer entre le *read* et le génome de référence comme l'illustrent les bases en gras.

Le nombre moyen de *reads* alignés par position du génome est appelé la couverture de séquençage. Plus elle est grande, plus le séquençage est complet. Le nombre de *reads* alignés pour un échantillon expérimental est appelé la profondeur de séquençage.

L'algorithme d'alignement peut également s'appuyer sur une annotation du génome déjà connue (*i.e.* la liste des positions de chaque gène sur le génome). L'annotation déjà connue d'un génome donné peut être téléchargée sous la forme de fichiers `.GFF` à partir des bases de données publiques d'annotation, sur le site NCBI ou `Ensembl` (Biomart)[4]. L'étape d'alignement ne requiert pas cette annotation du génome et peut être effectuée sans cette information.

De nombreuses méthodes existent pour effectuer cet alignement, par exemple `TopHat` (Trapnell et al., 2009). Les programmes d'alignement transforment les fichiers de *reads* bruts `.FASTQ` en fichier binaire de format `BAM` ou texte de format `SAM` signifiant *Sequence Alignment/Map*. Les fichiers `.SAM` et `.BAM` contiennent l'information de position de chaque *read* sur le génome. Ils sont volumineux et doivent être manipulés à l'aide d'outils spécifiques, par exemple `samtools` (Li et al., 2009).

4. **Comptage des *reads* :** Une fois la position de chaque *read* connue, les *reads* alignés sur chaque région génomique d'intérêt sont dénombrés. L'annotation de référence téléchargée à partir des bases de données publiques permet de connaître la délimitation des régions génomiques d'intérêt (en particulier, la position des gènes). Cependant, les programmes permettent aussi de détecter de nouvelles régions génomiques encore non identifiées. Les régions génomiques nouvellement détectées viennent ainsi enrichir les bases de données existantes. De nombreux programmes existent également pour compter l'abondance des transcrits par région génomique, en particulier `Cufflink` (Trapnell et al., 2010) ou `HTSeq-counts` (Anders et al., 2014) . Le fichier produit en sortie de cette étape d'estimation d'abondance des *reads* par région génomique est un tableau de comptage, comme l'illustre le Tableau 1.1.

La technologie RNA-seq vient révolutionner la mesure de l'expression des gènes. En s'appuyant sur la puissance de calcul à disposition (notamment pour l'étape d'alignement des *reads*), elle permet la mesure *haut-débit* de l'expression simultanée de plusieurs milliers de gènes à la fois, contrairement à la technologie SAGE dont le débit est beaucoup moins important. De plus, le RNA-seq mesure l'expression sans limite d'amplitude :

4. http ://www.ensembl.org

	gène 1	gène 2	gène 3	gène 4	gène 5	gène 6	...
échantillon 1	4	19	2987	0	65	1905	...
échantillon 2	0	18	1206	0	29	121	...
échantillon 3	6	20	12	48	299	169	...

TABLE 1.1 – Tableau de comptages résumant l'abondance des *reads* par région génomique d'intérêt pour trois réplicats biologiques.

un gène peut avoir un alignement de 0 *read* ou plus de 100 000 *reads*, contrairement à la technologie des puces à ADN. L'alignement des séquences peut être effectué sur un génome de référence potentiellement incomplet (alignement *de novo*) ce qui permet la découverte de nouveaux gènes, ce qui n'est pas possible avec les données de puces à ADN. En particulier, cette technologie permet d'examiner l'expression spécifique aux allèles, qui sont les variantes d'un même gène résultant d'une mutation et ayant la même fonction que le gène initial selon ses modalités propres. La technologie RNA-seq offre ainsi un niveau de précision supérieur à celui des données de puces à ADN.

De nombreuses méthodes statistiques sont utiles à différentes étapes de mesure de l'expression des gènes par la technologie RNA-seq : contrôle qualité des *reads*, alignement des *reads* et estimation de l'abondance des transcrits. Dans cette thèse, on s'intéressera uniquement aux méthodes statistiques destinées à analyser le tableau de comptages (Table 1.1) après estimation de l'abondance des *reads*.

1.2 Modélisation statistique des données RNA-seq

1.2.1 Modélisation et normalisation

Les notations utilisées tout au long de la thèse sont fixées ici. Par abus de langage, toute région génomique d'intérêt est assimilée à un gène. On utilise également le terme d'échantillon pour désigner les réplicats biologiques ou techniques. Les réplicats techniques désignent les échantillons collectés sur un même individu pour une même condition expérimentale. Les réplicats biologiques désignent les échantillons collectés sur des individus différents. Les jeux de données analysés dans cette thèse ne contiennent pas de réplicat technique : il s'agit toujours de réplicats biologiques d'une même condition ou de conditions expérimentales différentes. On utilise le terme d'échantillon pour désigner les réplicats biologiques et l'on précise les différentes conditions expérimentales si besoin. Le tableau de comptage d'expression de gènes (exemple : tableau 1.1) est représenté par une matrice, notée $\mathbf{y}$. Cette matrice d'expression est de taille $(n \times p)$, n est le nombre d'échantillons séquencés et p le nombre de gènes. L'entrée y_{ij} désigne l'expression du gène j pour l'échantillon i. Pour tout échantillon $i \in 1, \ldots, n$, le vecteur $\mathbf{y}_i$ est le vecteur d'expression de l'échantillon i pour les p gènes. Pour tout gène $j \in 1, \ldots, p$, le vecteur $\mathbf{y}^j$ est le vecteur d'expression du gène j pour les n échantillons. Si les échantillons correspondent à différentes conditions expérimentales, cette condition expérimentale est notée $c(i)$ pour chaque échantillon i.

Modélisation des données RNA-seq

Le nombre de *reads* alignés pour un échantillon donné est fortement dépendant de la profondeur de séquençage qui est une contrainte spécifique de la technologie RNA-seq. Pour un échantillon donné, ce nombre peut être considéré fixe. Sachant ce nombre $N_i = \sum_{j=1}^{p} y_{ij}$, le vecteur aléatoire $\mathbf{Y}_i$ modélisant la distribution des *reads* sur les p régions génomiques du génome pour l'échantillon i peut être modélisé par une loi multinomiale de paramètres N_i et $(\pi_{i1}, \ldots, \pi_{ip})$:

$$p(Y_{i1} = y_{i1}, \ldots, Y_{ip} = y_{ip}) = \frac{N_i}{y_{i1}! \ldots y_{ip}!} \pi_{i1}^{y_{i1}} \ldots \pi_{ip}^{y_{ip}}.$$

Chaque composante Y_{ij} de ce vecteur multinomial $\mathbf{Y}_i$ suit une loi binomiale de paramètres N_i et π_j. Le nombre de *reads* alignés par échantillon N_i est élevé et π_j, la proportion de *reads* alignés sur le gène j est petite. L'approximation d'une loi binomiale par une loi de Poisson est applicable :

$$\begin{aligned} Y_{ij} &\sim \mathcal{P}(\lambda_{ij}), \\ p(Y_{ij} = y_{ij}) &= \frac{\lambda_{ij}^{y_{ij}}}{y_{ij}!} e^{-\lambda_{ij}}, \\ E(Y_{ij}) &= \lambda_{ij}, \\ V(Y_{ij}) &= \lambda_{ij}. \end{aligned}$$

Le modèle de Poisson est utile en particulier pour l'analyse de réplicats techniques d'une même condition (Marioni et al., 2008). Cependant, il modélise mal la grande variabilité inter-échantillons observée dans les données RNA-seq lorsque les différents réplicats biologiques sont des réplicats biologiques et non techniques d'une même condition expérimentale. Afin de prendre en compte cette grande variabilité, des modèles de loi de Poisson surdispersée ou de loi binomiale négative ont été proposés comme solutions alternatives au modèle de Poisson : $Y_{ij} \sim \mathrm{NB}(\lambda_{ij}, \phi_j)$. Le paramètre λ_{ij} est modélisé de manière appropriée et $\phi_j \geq 0$ est le paramètre de dispersion du gène j. Plusieurs paramétrisations de la loi binomiale négative sont possibles, mais il est courant d'adopter la convention suivante :

$$\begin{aligned} Y_{ij} &\sim \mathrm{NB}(\lambda_{ij}, \phi_j), \\ p(Y_{ij} = y_{ij}) &= \left(\frac{\phi_j}{\phi_j + \lambda_{ij}}\right)^{\phi_j} \frac{\Gamma(\phi_j + y_{ij})}{y_{ij}!\,\Gamma(\phi_j)} \left(\frac{\lambda_{ij}}{\phi_j + \lambda_{ij}}\right)^{y_{ij}}, \\ E(Y_{ij}) &= \lambda_{ij}, \\ V(Y_{ij}) &= \lambda_{ij} + \lambda_{ij}^2 \phi_j. \end{aligned}$$

Que ce soit pour la loi de Poisson ou la loi binomiale négative, il convient de modéliser le paramètre λ_{ij} de manière appropriée, par exemple en supposant que $\lambda_{ij} = e_i g_j$ où g_j est un paramètre spécifique au gène j et e_i est un paramètre spécifique à l'échantillon i. Ces modélisations sont adaptées en fonction du type d'analyse effectuée : analyse

différentielle, classification ou inférence de réseaux. Les paramètres doivent prendre en compte les biais techniques de la technologie RNA-seq que nous détaillons dans la section suivante.

Prise en compte de la profondeur de séquençage par échantillon

Le nombre de *reads* alignés pour un gène et un échantillon donné est une mesure relative et non absolue de l'expression du gène. Ce nombre dépend du nombre total de fragments d'ADN alignés pour l'échantillon donné comme l'illustre l'exemple du tableau 1.2.

Comptages bruts

	gène 1	gène 2	gène 3	...	nombre total de reads
échantillon 1	10	10	10	...	1000
échantillon 2	100	100	100	...	10000

Comptages divisés par le nombre total de *reads* de l'échantillon

	gène 1	gène 2	gène 3	...	nombre total de reads
échantillon 1	0.01	0.01	0.01	...	1000
échantillon 2	0.01	0.01	0.01	...	10000

TABLE 1.2 – Tableau de comptages résumant l'abondance des *reads* par région génomique d'intérêt pour deux échantillons (haut). Tableau de comptages normalisés par le nombre total de *reads* alignés pour chaque échantillon (bas).

Pour que l'expression d'un gène soit comparable entre plusieurs réplicats biologiques, il convient de prendre en compte le nombre total de *reads* alignés pour chaque échantillon, aussi appelé taille de librairie ou profondeur de séquençage par échantillon. Le comptage par million (*count per million* CPM) du gène j pour l'échantillon i correspond au nombre de reads y_{ij} alignés sur cette région, normalisé par le nombre total de reads alignés pour l'échantillon i (noté $N_i = \sum_{j=1}^{p} y_{ij}$) divisé par un million :

$$\text{cpm}(y_{ij}) = \frac{y_{ij}}{N_i/10^6}.$$

Le log-comptage par million (*log-count per million* log-cpm) du gène j pour l'échantillon i a été utilisé par Law et al. (2014). Le log-cpm correspond au logarithme du nombre de *reads* y_{ij} alignés sur cette région, normalisé par le nombre total de lectures alignées pour l'échantillon i exprimé en un million :

$$\text{log-cpm}(y_{ij}) = \log_2\left(\frac{y_{ij} + 0.5}{(N_i + 1)/10^6}\right).$$

Prise en compte de la longueur du gène

Pour comparer l'expression de différents gènes au sein d'un échantillon donné, il convient de prendre en compte la longueur de la région génomique sur laquelle les

fragments sont alignés. En effet, la longueur d'un gène peut varier entre 400 et 2 millions de paires de bases. La longueur du gène peut donc expliquer à elle seule les différences d'expression observées. Pour un gène j, on suppose cette longueur connue. La longueur du gène, notée L_j et exprimée en nombre de paires de bases, ne peut être estimée à partir du simple tableau de comptages RNA-seq. Elle doit être extraite des bases de données publiques d'annotation du génome. La méthode de normalisation la plus simple pour la comparaison de mesure d'expression entre deux gènes consiste à diviser chaque comptage par la longueur du gène L_j correspondante. On obtient ainsi une mesure d'expression du gène par paire de bases. Cependant, cette mesure d'expression ne prend pas en compte les différences de taille de librairie. La prise en compte des différences de taille de librairie est réalisée dans le calcul du nombre de *transcripts per million* (TPM) (Li et al., 2009) ou dans le calcul du nombre de lectures par kilobase d'exon par million de lectures alignées (*reads per kilobase of exon per million reads mapped*, RPKM) (Mortazavi et al., 2008) :

$$\text{rpkm}(y_{ij}) = \frac{y_{ij}}{\left(\frac{L_j}{10^3}\right)\left(\frac{N_i}{10^6}\right)}.$$

Autres biais des données RNA-seq et modélisation

D'autres biais des données RNA-seq existent. Un exemple est le biais GC. Le contenu en GC d'un gène désigne la proportion de nucléotides G et C dans la séquence du gène. Si le contenu en GC est sous-représenté ou sur-représenté, le séquençage du *read* est plus difficile, ce qui peut être pris en compte dans le calcul d'unité d'expression des données RNA-seq. Risso et al. (2011); Hansen et al. (2012) ajustent à l'aide d'une régression loess les log-comptages sur le contenu en GC de chaque gène.

Dans cette thèse, nous ne présentons pas toutes les méthodes possibles de correction des biais des données RNA-seq. De plus, ces méthodes dépendent fortement du type d'analyse statistique effectuée en aval. On peut cependant distinguer deux grandes approches de modélisation des données RNA-seq :

1. Les modèles discrets (loi de Poisson ou loi binomiale négative) qui modélisent directement les comptages des *reads* et prennent en compte les biais des données RNA-seq directement dans l'écriture des paramètres du modèle,
2. Les modèles gaussiens qui modélisent les comptages normalisés et prennent en compte les biais des données RNA-seq préalablement à l'utilisation du modèle.

1.2.2 Analyse différentielle

L'objectif de l'analyse différentielle est de détecter les gènes différentiellement exprimés entre différentes conditions expérimentales testées. Pour que l'analyse différentielle soit pertinente, il convient de planifier l'expérience avec soin, et de choisir, en fonction du budget disponible, la profondeur de séquençage (le nombre de *reads* séquencés par échantillon) et le nombre d'échantillons à séquencer. Busby et al. (2013) ont formulé quelques recommandations à ce sujet. L'analyse différentielle se décompose en plusieurs étapes :

1. **Définition des hypothèses à tester :** Pour chaque gène j, l'analyse différentielle détermine si une différence d'expression est observée entre deux ou plusieurs conditions expérimentales à l'aide de tests d'hypothèses. Pour simplifier les notations, on considère la comparaison de deux conditions expérimentales uniquement : condition 1 et condition 2. Le test d'hypothèse détermine s'il existe une différence entre la moyenne λ_{1j} d'expression du gène j dans la condition 1 et la moyenne λ_{2j} d'expression du gène j dans la condition 2 :

$$H_{0j} = \{\lambda_{1j} = \lambda_{2j}\},$$

vs.

$$H_{1j} = \{\lambda_{1j} \neq \lambda_{2j}\}.$$

2. **Choix de modèle, estimation des paramètres et test d'hypothèse :** Afin d'établir ces tests d'hypothèses, il convient de modéliser l'expression des gènes dans chaque condition par une variable aléatoire et d'effectuer un test de comparaison sur le paramètre de la distribution de la variable aléatoire à l'aide d'un test exact si possible, test du rapport de vraisemblance ou test de Wald pour tester la différence d'expression du gène en question.
3. **Correction pour les tests multiples :** Le test d'hypothèse est effectué pour chaque gène. On dispose d'un vecteur de p-values correspondant de longueur p. Puisque le nombre de gènes est élevé, il convient de corriger le seuil de rejet de l'hypothèse nulle afin de contrôler correctement le nombre de gènes détectés différentiellement exprimés à tort, aussi appelé le nombre de faux positifs ou *False Discovery Rate* (FDR). Différentes méthodes existent pour contrôler le nombre de faux positifs, correspondant à différentes manières de mesurer la quantité de faux positifs. Si l'on souhaite contrôler le nombre de faux positifs en probabilité, on adoptera la procédure de Bonferroni améliorée par Hochberg (1988). Si l'on souhaite contrôler le taux de faux positifs en espérance, on adopte la procédure Benjamini and Hochberg (1995). La procédure de Benjamini-Yekutieli est une adaptation de la procédure de Benjamini et Hochberg dans le cadre du contrôle du taux de faux positifs avec hypothèse de dépendance entre les différents tests.

L'étape 2 doit être adaptée aux données RNA-seq. Pour les données microarray qui sont continues, il est possible de construire des modèles d'analyse statistique en se basant sur des hypothèses de normalité des données (Smyth, 2004). Ces techniques d'analyse, adaptées aux données gaussiennes ne peuvent pas être appliquées directement aux données RNA-seq qui sont des données de comptage, discrètes et positives. De nombreux travaux de recherche ont été effectués sur ce sujet.

Plusieurs modèles sont possibles, comme des lois de Poisson (Wang et al., 2010; Marioni et al., 2008), des lois de Poisson sur-dispersées (Auer and Doerge, 2011), des lois binomiales négatives (Anders and Huber, 2010; Robinson and Oshlack, 2010) ou des lois normales (Law et al., 2014).

Le modèle proposé par Love et al. (2014) et utilisé dans le package `DESeq2` est un modèle linéaire généralisé :

$$Y_{ij} \sim \mathrm{NB}(\mu_{ij}, \phi_j),$$
$$\mu_{ij} = s_i \lambda_{ij},$$

$$log_2(\lambda_{ij}) = D_{i.}\beta_j.$$

Le comptage Y_{ij} du gène j pour l'échantillon i est modélisé à l'aide d'une loi binomiale négative de moyenne μ_{ij} et de paramètre de dispersion ϕ_j spécifique au gène j. La moyenne ajustée μ_{ij} est composée d'un paramètre spécifique à la taille de la librairie s_i et d'un paramètre λ_{ij} proportionnel à l'expression du gène j dans l'échantillon i. Le vecteur coefficient β_j modélise les variations de l'expression du gène j en fonction des conditions expérimentales de chaque échantillon résumées dans la matrice de design D.

Cette modélisation prend en compte la variabilité de la profondeur de séquençage pour les différents réplicats biologiques. Il existe différentes façons de estimer les facteurs de normalisation pour les différences de profondeur de séquençage entre échantillons, par exemple en ne considérant qu'un sous-ensemble de gènes non-différentiellement exprimés. En effet, les gènes différentiellement exprimés entre différentes conditions présentent généralement des comptages très déséquilibrés entre échantillons qui viennent fausser l'estimation de la profondeur de séquençage. Les deux méthodes de normalisation correspondantes les plus utilisées sont la méthode *Trimmed M-Means* (TMM)(Robinson and Oshlack, 2010) et la méthode *median-of-ratios* utilisée dans les packages `DESeq` et `DESeq2` (Anders and Huber, 2010). Ces deux méthodes sont particulièrement appréciées pour corriger le biais induit par des tailles de librairie très différentes au sein d'un même jeu de données (Dillies et al., 2013). Concernant la prise en compte de la longueur du gène, il a été montré que l'utilisation des RPKMs réduisait la puissance de détection des gènes différentiellement exprimés (Oshlack and Wakefield, 2009; Dillies et al., 2013).

Une fois le modèle choisi, il convient d'estimer les paramètres des modèles correspondants. Pour le modèle de loi binomiale négative, les deux méthodes les plus utilisées ont été proposées par Robinson and Oshlack (2010) et Love et al. (2014) et sont disponibles respectivement dans les packages R `edgeR` et `DESeq2`. L'originalité de ces méthodes réside dans l'estimation du paramètre de dispersion de la loi binomiale négative. Les données RNA-seq ont généralement peu de réplicats biologiques et le paramètre de dispersion est de ce fait mal estimé. Les méthodes utilisent des techniques de partage d'information entre les gènes afin d'estimer le paramètre de dispersion de chaque gène. Le package `DESeq2` modélise la tendance moyenne-variance afin d'estimer ce paramètre de dispersion, tandis que le package `edgeR` utilise un compromis entre une dispersion commune à tous les gènes et une dispersion spécifique à chaque gène. On parle alors de shrinkage du paramètre de dispersion vers une dispersion commune.

Les contributions méthodologiques à l'analyse différentielle des données RNA-seq sont nombreuses et font l'objet de recherches actives (Oshlack et al., 2010; Dillies et al., 2013). Dans cette thèse, nous ne nous intéressons pas à l'analyse différentielle en tant que telle. Nous avons cependant proposé une méthode de filtrage des données RNA-seq qui est une étape préliminaire à l'analyse différentielle. Cette contribution est motivée par la nécessité de réduire la taille du jeu de données afin d'améliorer la puissance de détection des tests d'analyse différentielle décrits plus haut.

1.2.3 Détection de gènes co-exprimés

Au-delà de l'analyse différentielle des données d'expression, d'autres types d'analyses peuvent être effectués afin de mieux comprendre les processus sous-jacents aux données

observées et de formuler des hypothèses biologiques pour des recherches futures. L'une d'entre elles est la détection de gènes co-exprimés. Nous détaillons le principe de cette analyse, les différentes méthodes utilisées en pratique et les problèmes auxquels nous nous intéressons.

L'analyse de co-expression des données d'expression de gènes consiste à grouper les gènes ayant des profils d'expression similaires à travers différentes conditions. Elle est particulièrement intéressante dans le cas où plus de deux conditions expérimentales sont étudiées et permet d'aller plus loin que l'analyse différentielle dans l'étude du transcriptome.

Plusieurs techniques de classification non supervisée permettent de classer les gènes en fonction de leur profil d'expression : la classification hiérarchique (Ward, 1963), la méthode des k-means (MacQueen, 1967), les réseaux de neurones (Tamayo et al., 1999) ou les modèles de mélange (McLachlan and Peel, 2000). Pour l'analyse de co-expression des données d'expression de puces à ADN, Yeung et al. (2001) ont montré l'intérêt des modèles de mélange. Nous adoptons le même point de vue pour l'analyse de co-expression des données RNA-seq.

L'objectif de l'analyse de co-expression est l'enrichissement des bases de données d'*annotation fonctionnelle* des gènes. Généralement, les gènes co-exprimés sont impliqués dans les mêmes processus biologiques (Eisen et al., 1998; Jiang et al., 2004). Ces groupes de gènes co-exprimés permettent de formuler des hypothèses sur les fonctions possibles des gènes : on peut ainsi inférer les fonctions d'un gène orphelin, dont le rôle n'a pas encore été identifié, en regardant les fonctions déjà connues des gènes classés dans le même groupe de co-expression.

Les méthodes de classification utilisées en pratique sur les données RNA-seq

En classification des données RNA-seq, certains auteurs classent les réplicats biologiques et non les gènes. Plusieurs méthodes ont été utilisées en pratique : clustering hiérarchique avec la distance euclidienne sur les données RNA-seq après transformation pour stabiliser la variance (*Variance stabilizing transformation*, VST) (Anders and Huber, 2010), clustering hiérarchique avec la distance de corrélation de Pearson sur les données RNA-seq exprimées en RPKM (Severin et al., 2010) ou classification hiérarchique avec une distance entre réplicats biologiques calculée à partir d'un modèle de Poisson log linéaire (Witten, 2011). On remarque que deux stratégies existent : soit les auteurs transforment les données pour utiliser les méthodes classiques de classification (Anders and Huber, 2010; Severin et al., 2010), soit les auteurs proposent un modèle adapté aux données RNA-seq (Witten, 2011).

Le constat est identique pour la classification des gènes, Li et al. (2009) utilisent un algorithme de type k-means sur les données RNA-seq log-transformées, mais des techniques adaptées au caractère discret des données RNA-seq existent. Une méthode de classification de type k-means couplée à une distance entre gènes basée sur un modèle log-linéaire de Poisson a été proposée pour la classification des *tags* de la technologie SAGE, ancêtre du RNA-seq (Huang et al., 2008). Plus récemment, deux modèles spécifiques aux données RNA-seq ont été proposés : un modèle de mélange de loi de Poisson (Rau et al., 2015) et un modèle de mélange de lois binomiales négatives (Si et al., 2013).

Problématique abordée : Le choix de modélisation est une question importante et aucune méthode de classification n'est clairement établie en analyse de co-expression des données RNA-seq. Dans le cadre de la classification par modèle de mélange, nous soulevons la question du choix de modélisation (discrète ou continue) des données RNA-seq.

Les bases de données d'annotation fonctionnelle des gènes

Le terme *annotation fonctionnelle* désigne l'ensemble des fonctions que l'on peut associer aux produits du gène (protéines ou ARN). Plus généralement, le terme d'annotation du génome se réfère à l'ensemble de méta-données associées aux régions codantes du génome, en particulier, la position sur le génome de la région codante des gènes. Dans cette thèse, le terme annotation désignera l'annotation fonctionnelle et non l'annotation au sens général si aucune précision n'est donnée. Plusieurs bases de données ont été construites à partir des références bibliographiques collectées. La plus connue est la base de données Gene Ontology (GO) (Ashburner et al., 2000), construite sous la forme de graphe acyclique orienté hiérarchisant les termes d'annotations. D'autres bases de données ont été créées, comme la base de données Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000). Plus récemment, la base de données MSigDB (Molecular Signatures databases) (Liberzon et al., 2011) a regroupé et facilité l'utilisation de cette information en proposant notamment des listes de fonctions filtrées manuellement. Ces bases de données contiennent des informations riches sur un large spectre d'espèces (e.g., humain, souris, mouche, *Arabidopsis thaliana*). Cependant, notre connaissance des annotations fonctionnelles des gènes est loin d'être complète (Tipney and Hunter, 2010). Les analyses de co-expression permettent de compléter ces bases de données.

Problématique abordée : Nous verrons comment les informations déjà existantes dans les bases de données peuvent contribuer à améliorer la qualité de l'analyse de co-expression et la pertinence des groupes de gènes détectés.

1.2.4 Inférence de réseaux de régulation de gènes

L'objectif de l'analyse différentielle des données d'expression de gènes ou la classification de ces données par modèle de mélange n'est pas de modéliser la dépendance entre les gènes. Or, la biologie des systèmes s'intéresse aux interactions entre les différents acteurs du système biologique. Les gènes interagissent entre eux, le plus souvent par l'intermédiaire de produits facteurs de transcription. Les facteurs de transcription sont des protéines produites par des gènes qui viennent réguler ou initier la transcription d'autres gènes. Cependant, mesurer la quantité de protéines présentes dans la cellule est très difficile du fait de l'instabilité des protéines qui se dégradent très rapidement. On préfère mesurer la quantité de transcrits et considérer que cette quantité est représentative du niveau d'expression du gène (i.e. du niveau de protéines produites). Des variations simultanées de ces qualités d'expression pour deux gènes peuvent ainsi être synonymes d'interaction entre ces deux gènes.
Un graphe, ensemble de nœuds et ensemble d'arêtes joignant ces nœuds, formalise ces interactions et permet de les visualiser. Dans le cas particulier des réseaux de régula-

tion de gènes, les nœuds représentent les gènes et les arêtes représentent les interactions entre les gènes. Les interactions entre les gènes peuvent prendre de multiples formes et ne peuvent se résumer à un seul type d'interaction. Le formalisme de graphe n'est donc qu'une simplification de la réalité utile pour formuler des hypothèses sur le rôle des gènes. Par exemple, la détection de modules de gènes très connectés dans le réseau peut éventuellement permettre de découvrir de nouveaux gènes d'intérêt ou de nouvelles voies biologiques (*biological pathways*).

D'un point de vue méthodologique, il existe de nombreuses manières de reconstituer un graphe. La méthode la plus simple pour inférer un réseau de régulation de gènes à partir de données d'expression consiste à calculer les corrélations simples paire à paire entre les gènes et à fixer un seuil en dessous duquel les corrélations sont considérées comme insignifiantes afin de déterminer l'absence d'arête dans le graphe. Cette méthode d'inférence de réseaux est très populaire en analyses des données d'expression (Butte et al., 2000; De la Fuente et al., 2004). La méthode *Weighted Correlation Network Analysis* (WGCNA), proposé par Langfelder and Horvath (2008) est également très utilisée, et implémentée dans le package R `WGCNA`. Cette méthode d'inférence est basée sur le feuillage de la matrice de covariance empirique. Des méthodes basées sur le calcul de l'information mutuelle ont également été proposée (Basso et al., 2005; Margolin et al., 2006). Une implémentation de méthode d'inférence de réseaux basée sur l'information mutuelle est disponible dans le package R/Bioconductor `minet` (Meyer et al., 2008). Cependant, une corrélation très élevée entre le niveau d'expression de deux gènes n'implique pas un lien *direct* entre les deux gènes. Reverter and Chan (2008) ont proposé une méthode combinant le calcul de corrélation partielle et d'information mutuelle (*Partial Correlation Information Theory* (PCTI) algorithm) pour supprimer les liens non *directs* entre les gènes du graphe. Afin de ne détecter que les liens *directs* entre les gènes, de nombreuses méthodes basées sur l'utilisation du modèle graphique gaussien ont été utilisées (Schäfer and Strimmer, 2005; Peng et al., 2009). Toutes les méthodes décrites précédemment permettent de reconstituer des réseaux de dépendance non orientés. Les réseaux bayésiens permettent d'inférer des graphes orientés (Chen et al., 2006) et sont particulièrement utiles sur les données d'intervention (*knock-out*). La modélisation des données d'expression par des équations différentielles ordinaires ou stochastiques permet également de reconstituer des réseaux de gènes. Ces méthodes sont particulièrement intéressantes en analyse des données cinétiques.

Des comparaisons entre les différentes méthodes d'inférence de réseaux sont proposées par Werhli et al. (2006) et Allen et al. (2012). Il serait cependant difficile de hiérarchiser les méthodes d'inférence de réseaux. La dépendance existant entre deux gènes peut prendre de multiples formes et ne peut être résumée à un simple indice statistique. Combiner les résultats des différentes méthodes semble être l'approche la plus informative (Novère, 2015). Dans cette thèse, nous focalisons notre attention sur les modèles graphiques non orientés et l'inférence de réseaux de dépendance *directe* entre les gènes.

Les méthodes d'inférence utilisées en pratique sur les données RNA-seq

Pour inférer des réseaux de gènes à partir des données RNA-seq, Giorgi et al. (2013) ont utilisé le coefficient de corrélation de Pearson calculé sur les données RNA-seq transformées pour stabiliser la variance (*Variance Stabilizing Transformation*, VST). Hong

et al. (2013) ont proposé une méthode basée sur le calcul d'un coefficient de corrélation canonique. Iancu et al. (2012) ont utilisé la méthode WGCNA sur les données RNA-seq normalisées et transformées pour stabiliser la variance (VST). Ces méthodes appliquées sur les données RNA-seq sont donc principalement des méthodes d'inférence de réseaux de dépendances marginales : le réseau est reconstitué à partir d'un indice calculé marginalement sur toutes les paires de gènes possibles. Certaines méthodes visent cependant à reconstituer des réseaux de dépendances conditionnelles, où la dépendance entre deux gènes est calculée conditionnellement aux valeurs prises par les autres gènes. Un exemple est l'utilisation du modèle graphique gaussien sur les données RNA-seq exprimées en RPKM par Cai et al. (2012). Plus récemment, Allen and Liu (2013) ont proposé un modèle log-linéaire de Poisson, adapté aux données RNA-seq après transformation. Notons également que ces méthodes d'inférence de réseaux nécessitent un nombre d'échantillons relativement élevé.

Problématiques abordées : Comme en analyse différentielle et en analyse de co-expression, le choix de modélisation des données RNA-seq (discrète ou continue) est une question importante en inférence de réseaux. Comme l'analyse différentielle, l'inférence de réseaux souffre également du faible nombre d'échantillons à disposition. Nous proposerons également une solution palliative à ce problème.

Chapitre 2

Cadre statistique

Pour chaque modèle statistique, nous distinguons les unités statistiques, indexées par l'indice $i = 1, \ldots, n$, et les variables, indexées par l'indice $j = 1, \ldots, p$.

2.1 Classification

2.1.1 Les méthodes de classification

On considère une matrice $\mathbf{y}$ de données d'observations de taille $n \times p$. On souhaite classer les n observations, $\mathbf{y}_i$ pour $i = 1, \ldots, n$, en fonction des p variables $\mathbf{y}^j$ pour $j = 1, \ldots, p$. Plusieurs techniques de classification non supervisée permettent de classer des observations en fonction de variables.

La classification hiérarchique
La classification hiérarchique requiert le choix d'une métrique et d'un critère d'agrégation. Deux exemples de métriques sont la distance euclidienne ou le coefficient de corrélation de Pearson entre deux observations. Partant d'une partition en singleton des observations, le critère d'agrégation détermine la manière dont on agrège les observations pour former les classes (*single linkage, average linkage, complete linkage*). La hiérarchie des classes inférées peut être représentée à l'aide d'un dendogramme. L'utilisateur doit alors fixer la profondeur qu'il désire dans le dendogramme pour obtenir une classification des observations.

La méthode des k-means
Cette méthode consiste à minimiser l'inertie intra-classe (la distance entre les observations d'une même classe) et à maximiser l'inertie inter-classe (la distance entre les observations appartenant à des classes différentes). Cette méthode est sensible aux valeurs extrêmes.

Les modèles de mélange
Ils offrent une définition rigoureuse de la notion de classes, les observations appartenant à une même classe sont des échantillons issus de variables aléatoires suivant la même loi. Les données observées sont donc la réalisation de variables aléatoires de différentes lois. Une présentation détaillée des modèles de mélange est proposée par McLachlan and Peel (2000).

Cette liste n'est pas exhaustive. D'autres méthodes existent pour classer des observations comme les réseaux de neurones (Tamayo et al., 1999), les méthodes basées sur la théorie des graphes ou les machines à vecteurs de support. Nous concentrons notre attention sur les modèles de mélange car ils offrent une définition rigoureuse de la notion de classes et un cadre probabiliste facilitant le choix de modèle.

2.1.2 Les modèles de mélange

On suppose qu'il existe une partition des observations $\mathbf{y}_1, \ldots, \mathbf{y}_n$ en K classes $C_1, \ldots, C_K$. Cette partition est notée $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$ où $\mathbf{z}_i = (z_{i1}, \ldots, z_{iK})$ est un vecteur binaire indiquant l'appartenance de l'observation i aux K classes : $z_{ik} = 1$ si l'observation $\mathbf{y}_i$ appartient à la classe k, et $z_{ik} = 0$ sinon. Cette partition $\mathbf{z}$ est une

donnée non observée et l'objectif de l'analyse est l'estimation de cette partition $\mathbf{z}$ à partir des données observées $\mathbf{y}$. Pour tout i, on suppose que l'observation $\mathbf{y}_i$ appartient à la classe C_k si elle est la réalisation d'une variable aléatoire $\mathbf{Y}_i$ conditionnellement à l'événement $Z_{ik} = 1$ où $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{iK})$ est le vecteur indiquant l'appartenance de la variable aléatoire $\mathbf{Y}_i$ aux classes : $Z_{ik} = 1$ si $\mathbf{Y}_i$ appartient à la classe k, $Z_{ik} = 0$ sinon. La distribution de la variable $\mathbf{Y}_i$ conditionnellement à l'appartenance à la classe k suit une loi de densité $f_k(\mathbf{y}_i)$. La variable $\mathbf{Z}_i$ suit une loi multinomiale de paramètre $(1, \mathbf{p})$, où $\mathbf{p} = (p_1, \ldots, p_K)$ est le vecteur des proportions de chaque composante du modèle de mélange tel que $p_k \in]0, 1[$ pour tout k et $\sum_{k=1}^{K} p_k = 1$. Chaque paramètre p_k de ce vecteur est égal à la probabilité que l'observation i provienne de la composante k : $p(Z_{ik} = 1)$. La densité jointe du couple $(\mathbf{Y}_i, \mathbf{Z}_i)$ s'écrit :

$$f(\mathbf{y}_i, \mathbf{z}_i) = \prod_{k=1}^{K} \left[p_k f_k(\mathbf{y}_i)\right]^{z_{ik}}.$$

La densité marginale de la variable aléatoire $\mathbf{Y}_i$ est un mélange de densité :

$$f(\mathbf{y}_i) = \sum_{k=1}^{K} p_k f_k(\mathbf{y}_i).$$

La distribution de la variable $\mathbf{Z}_i$ conditionnellement à l'évenement $\{Y_i = \mathbf{y}_i\}$ est une loi multinomiale de paramètre $(1, \mathbf{t}_i)$ où $\mathbf{t} = (t_{i1}, \ldots, t_{iK})$ est le vecteur des probabilités conditionnelles $t_{ik} = p(Z_{ik} = 1 \mid \mathbf{Y}_i = \mathbf{y}_i)$. Le théorème de Bayes conduit à la formulation suivante :

$$t_{ik} = \frac{p_k f_k(\mathbf{y}_i)}{\sum_{t=1}^{K} p_t f_t(\mathbf{y}_i)}.$$

Ces probabilités conditionnelles permettent d'attribuer une classe d'appartenance à l'observation i à l'aide de la règle du *maximum a posteriori* (MAP) :

$$z_{ik} = \begin{cases} 1 & \text{si } \arg\max_\ell t_{i\ell} = k, \\ 0 & \text{sinon.} \end{cases}$$

En pratique, ces probabilités conditionnelles ne sont pas connues car les proportions $(p_1, \ldots, p_K)$ et les densités de chaque composante du mélange $f_1(\mathbf{y}_i), \ldots, f_K(\mathbf{y}_i)$ ne sont pas connues. Elles doivent être estimées à partir des données observées $\mathbf{y}$. Afin de réaliser cette estimation, on suppose que les couples $(\mathbf{y}_1, \mathbf{z}_1), \ldots, (\mathbf{y}_n, \mathbf{z}_n)$ sont indépendants et identiquement distribués et qu'ils sont la réalisation des couples de variables aléatoires $(\mathbf{Y}_1, \mathbf{Z}_1), \ldots, (\mathbf{Y}_n, \mathbf{Z}_n)$. On suppose également que les densités $f_k(\mathbf{y}_i)$ s'écrivent sous la forme $f_k(\mathbf{y}_i; \mathbf{a}_k)$ et que les paramètres de chaque variable aléatoire $(\mathbf{a}_1, \ldots, \mathbf{a}_K)$ sont tous distincts. La densité de l'observation $\mathbf{y}_i$ s'écrit :

$$f(\mathbf{y}_i; K, \theta_K) = \sum_{k=1}^{K} p_k f_k(\mathbf{y}_i; \mathbf{a}_k).$$

Le vecteur des paramètres est $\theta_K = (p_1, \ldots, p_K, \mathbf{a}_1, \ldots, \mathbf{a}_K)$. Le choix de la distribution des variables aléatoires qui composent le mélange est important. Nous distinguons ici des modèles qui seront utilisés dans cette thèse.

Le modèle de mélange de lois gaussiennes

Le modèle de mélange de lois gaussiennes suppose que la densité des données pour une observation s'écrit :

$$f(\mathbf{y}_i; K, \theta_K) = \sum_{k=1}^{K} p_k \Phi(\mathbf{y}_i; \boldsymbol{\nu}_k, \boldsymbol{\Sigma}_k),$$

où $\Phi(\cdot)$ est la densité d'une loi normale multivariée de dimension p, de moyenne le vecteur $\boldsymbol{\nu}_k$ et de variance la matrice $\boldsymbol{\Sigma}_k$. Les modèles de mélange de lois gaussiennes sont identifiables à une permutation des composantes près : deux modèles de mélange ayant la même densité ont exactement les mêmes paramètres à une réindexation des composantes près. Ce n'est pas le cas des modèles de lois de Bernoulli, uniformes ou binomiales. Il existe plusieurs manières de décomposer la matrice de variance $\boldsymbol{\Sigma}_k$ (Celeux and Govaert, 1995) :

$$\boldsymbol{\Sigma}_k = L_k D_k A_k D_k^{-1}.$$

Le paramètre $L_k = \boldsymbol{\Sigma}^{\frac{1}{p}}$ caractérise le volume de la composante k. D_k est la matrice orthogonale des vecteurs propres de $\boldsymbol{\Sigma}_k$, elle indique l'orientation de la composante. Cette matrice modélise la dépendance entre les différentes variables $\mathbf{y}^1, \ldots, \mathbf{y}^p$: si elle est diagonale, les variables sont supposées indépendantes. La matrice diagonale A_k contient les valeurs propres de $\boldsymbol{\Sigma}_k$, normalisées et ordonnées par ordre décroissant. Cette matrice A_k indique la forme de la composante k. En imposant que l'un ou plusieurs de ces paramètres L_k, A_k ou D_k soient communs à toutes les composantes, on obtient une collection de 8 modèles de mélange gaussiens. Dans cette thèse, on utilise généralement le modèle $p_k L_k ADA$, noté $p_k L_k C$, où les proportions et les volumes des composantes sont libres, spécifiques à chaque composante et l'orientation et la forme des composantes sont communes à toutes les composantes du mélange.

Le modèle de mélange de lois de Poisson

Le modèle de mélange de lois de Poisson suppose que la densité des données pour une observation $\mathbf{y}_i$ s'écrit :

$$f(\mathbf{y}_i; K, \theta_K) = \sum_{k=1}^{K} p_k \prod_{j=1}^{p} \mathcal{P}(y_{ij}; \lambda_{jk}),$$

où $\mathcal{P}(\cdot; \lambda_{jk})$ dénote la loi de Poisson univariée de moyenne λ_{jk}. Ce modèle suppose l'indépendance conditionnelle des variables $\mathbf{y}^1, \ldots, \mathbf{y}^p$ sachant les classes. Cette hypothèse d'indépendance conditionnelle est forte et n'est généralement pas utilisée dans le cas des modèles de mélange de lois gaussiennes, pour lesquels la matrice $\boldsymbol{\Sigma}_k$ modélise la dépendance conditionnelle entre les variables. Cependant, l'implémentation d'une loi de Poisson mutivariée n'est pas simple (Karlis, 2003), et l'hypothèse d'indépendance facilite l'implémentation de l'estimation des paramètres.

Estimation des paramètres du modèles de mélange

De manière générale, la vraisemblance d'un modèle de mélange s'écrit :

$$\ell(\theta_K; \mathbf{y}) = \prod_{i=1}^{n} \sum_{k=1}^{K} p_k f_k(\mathbf{y}_i; \mathbf{a}_k) \, . \tag{2.1}$$

Il n'existe pas de forme explicite des estimateurs du maximum de vraisemblance des paramètres du modèle θ_K. Pour effectuer l'estimation des paramètres, on considère la vraisemblance complétée des données $(\mathbf{y}, \mathbf{z})$:

$$\ell(\theta_K; \mathbf{y}, \mathbf{z}) = \prod_{i=1}^{n} \prod_{k=1}^{K} (p_k f_k(\mathbf{y}_i; \mathbf{a}_k))^{z_{ik}}.$$

La partition $\mathbf{z}$ n'est pas observée. L'algorithme Espérance Maximisation, introduit par Dempster et al. (1977), permet l'estimation des paramètres d'un modèle comportant des variables latentes non observées. Après une étape d'initialisation des paramètres du modèle $\theta_K^{(0)}$, cet algorithme alterne deux étapes successives à chaque itération (b) :

Espérance : La première étape consiste à effectuer la maximisation de l'espérance de la log vraisemblance complétée conditionnellement aux observations $\mathbf{y}$ et aux paramètres courants estimés notés $\theta_K^{(b)}$:

$$\mathcal{Q}(\theta_K | \theta_K^{(b)}) = \mathbb{E}\left[\log \ell(\theta_K; \mathbf{y}, \mathbf{z}) | \mathbf{y}, \theta_K^{(b)}\right].$$

Pour les modèles de mélange, cette étape revient à calculer les probabilités conditionnelles, notées $t_{ik}^{(b)}$, que l'observation i appartient à la composante k sachant les données $\mathbf{y}$ et les paramètres courant du modèle $\theta_K^{(b)}$:

$$\begin{aligned} t_{ik}^{(b)} &= \mathbb{E}\left(Z_{ik} | \mathbf{y}, \theta_K^{(b)}\right), \\ &= \frac{p_k^{(b)} f_k(\mathbf{y}_i; \mathbf{a}_k^{(b)})}{\sum_{t=1}^{K} p_t^{(b)} f_t(\mathbf{y}_i; \mathbf{a}_t^{(b)})}. \end{aligned}$$

Maximisation : La deuxième étape de l'algorithme consiste à trouver $\theta_k^{(b+1)}$, les paramètres maximisant la quantité $\mathcal{Q}(\theta_K | \theta_K^{(b)})$ en θ_K. La maximisation de la quantité $\mathcal{Q}(\theta_K | \theta_K^{(b)})$ garantit d'augmenter la log vraisemblance $L(\theta_K | \mathbf{y})$.
Pour les proportions du modèle, cette étape de maximisation repose sur la formule suivante :

$$p_k^{(b+1)} = \frac{1}{n} \sum_{i=1}^{n} t_{ik}^{(b)} \, .$$

Dans le cas du modèle de mélange de lois gaussiennes, les paramètres de moyennes et de variances sont mis à jour de la manière suivante pour la forme $p_k L_k C$:

$$\boldsymbol{\nu}_k^{(b+1)} = \frac{\sum_{i=1}^{n} t_{ik}^{(b)} \mathbf{y}_i}{\sum_{i=1}^{n} t_{ik}^{(b)}},$$

$$\boldsymbol{\Sigma}_k^{(b+1)} = \frac{\sum_{i=1}^{n} t_{ik}^{(b)} (\mathbf{y}_i - \boldsymbol{\nu}_k^{(b)})(\mathbf{y}_i - \boldsymbol{\nu}_k^{(b)})^T}{\sum_{i=1}^{n} t_{ik}^{(b)}}.$$

Dans le cas du modèle de mélange de lois de Poisson, les paramètres de moyenne λ_{jk} pour tout $j \in 1, \ldots, p$ se mettent à jour de la manière suivante :

$$\lambda_{jk}^{(b)} = \frac{\sum_{i=1}^{n} t_{ik}^{(b)} y_{ij}}{\sum_{i=1}^{n} t_{ik}^{(b)} (\frac{1}{p} \sum_{j=1}^{p} y_{ij})}.$$

L'algorithme EM alterne les étapes d'**Espérance** et de **Maximisation** et met à jour les paramètres du modèle à chaque itération (b) jusqu'à convergence, parfois lente, vers un maximum local, qui n'est d'ailleurs pas forcement le maximum global de la fonction de vraisemblance. Diverses stratégies d'initialisation de l'algorithme ont été proposées pour améliorer la convergence de l'algorithme et la qualité de l'estimation, comme la stratégie du petit EM (Biernacki et al., 2003). Cette stratégie consiste à initialiser l'algorithme EM à l'aide d'un algorithme de classification de type $k-$means (MacQueen, 1967), puis à effectuer un petit nombre (par exemple 10) d'itérations de l'algorithme EM. Les paramètres estimés à l'issue de ce premier lancement de l'algorithme sont utilisés pour initialiser l'algorithme EM complet. Une fois l'estimation des paramètres effectuée $\hat{\theta}_K$, on peut déduire la partition des observations $\hat{\mathbf{z}}$ à l'aide de la règle du MAP :

$$\hat{z}_{ik} = \begin{cases} 1 & \text{si } \arg\max_\ell t_{i\ell}(\hat{\theta}_K) = k, \\ 0 & \text{sinon,} \end{cases}$$

$$t_{i\ell}(\hat{\theta}_K) = \frac{\hat{p}_\ell f_\ell(\mathbf{y}_i; \hat{\mathbf{a}}_\ell)}{\sum_{t=1}^{K} \hat{p}_t f_t(\mathbf{y}_i; \hat{\mathbf{a}}_t)}.$$

Sélection de modèle

Jusqu'à présent, nous avons considéré que le nombre de classes K était connu. Or, sur un jeu de données réels, ce nombres de classes n'estpas connu et nous devons le déduire à partir des données observées $\mathbf{y}$. Les modèles de mélange fournissent un cadre rigoureux pour choisir ce nombre de classes.

Le critère BIC

Une première solution au choix du nombre de classes K consiste à choisir le modèle qui maximise la vraisemblance intégrée :

$$f(\mathbf{y}; K) = \int_{\theta_K} f(\mathbf{y}; K, \theta_K) \pi(\theta_K) d\theta_K,$$

où $\pi(\theta_K)$ est une distribution a priori non informative sur les paramètres θ_K. Le calcul de cette vraisemblance intégrée est rarement possible. Le critère *Bayesian Information Criterion* (BIC), proposé par Schwarz (1978), est une approximation asymptotique du logarithme de la vraisemblance intégrée, réalisée à l'aide d'une approximation de Laplace (Lebarbier and Mary-Huard, 2006). Ce critère s'écrit :

$$\text{BIC}(K; \mathbf{y}) = \log f(\mathbf{y}; K, \hat{\theta}_K) - \frac{\nu_K}{2} \log(n),$$

où $\hat{\theta}_K$ est l'estimateur du maximum de vraisemblance des paramètres du modele de mélange et ν_K le degré de liberté du modèle à K composantes. Ce critère BIC fournit une approximation correcte de la vraisemblance lorsque le nombre d'observations n tend vers l'infini. Le critère BIC sélectionne le modèle le plus proche de la loi des données au sens de la pseudo-distance de Kullback–Leibler $KL(f, g)$ entre deux lois de probabilité de densités respectives f et g :

$$KL(f, g) = \int \log\left(\frac{f(x)}{g(x)}\right) f(x)dx.$$

Le critère ICL

Une alternative au critère BIC est le critère *Integrated Completed Likelihood criterion* (ICL) proposé par Biernacki et al. (2000) :

$$\text{ICL}(K; \mathbf{y}) = \text{BIC}(K; \mathbf{y}) - \text{Ent}(K; \mathbf{y}), \tag{2.2}$$

où $\text{Ent}(K; \mathbf{y})$ est un terme d'entropie qui mesure la confiance que l'on peut accorder à la classification :

$$\text{Ent}(K; \mathbf{y}) = -\sum_{i=1}^{n}\sum_{k=1}^{K} t_{ik}(\hat{\theta}_K) \log t_{ik}(\hat{\theta}_K) \geq 0. \tag{2.3}$$

En réalité, le critère ICL est une approximation du logarithme de la log-vraisemblance intégrée :

$$f(\mathbf{y}, \mathbf{z}; K) = \int_{\theta_K} f(\mathbf{y}, \mathbf{z}; K, \theta_K)\pi(\theta_K)d\theta_K.$$

Le terme d'entropie additionnel présent dans l'écriture du critère ICL (6.4) favorise les modèles qui présentent la partition des données la plus classifiante possible.

Les données représentées en Figure 2.1 sont simulées à partir d'un mélange de quatre lois gaussiennes bivariées représentées par les quatre différentes couleurs. Le critère ICL sélectionne une partition à trois classes. Les deux composantes en croix en haut à droite de la figure sont rassemblées en une seule classe. On obtient ainsi trois classes bien distinctes. À l'inverse, le critère BIC sélectionne le modèle fournissant la meilleure approximation de la densité des données. Il identifie correctement les composantes du modèle de mélange, et détecte bien quatre composantes. Cependant, les éléments appartenant aux deux composantes en croix (symbolisées par les triangles verts et les croix violettes) sont classés dans l'une ou l'autre des composantes avec incertitude et la confiance accordée à la classification en deux classes est faible.

2.2 Inférence de réseaux

2.2.1 Les méthodes d'inférence de réseaux

On considère une matrice $\mathbf{y}$ de taille $n \times p$ correspondant à n observations $\mathbf{y}_i$ pour $i = 1, \ldots, n$ et p variables $\mathbf{y}^j$ pour $j = 1, \ldots, p$. On suppose que les vecteurs $\mathbf{y}^1, \ldots, \mathbf{y}^p$ sont n

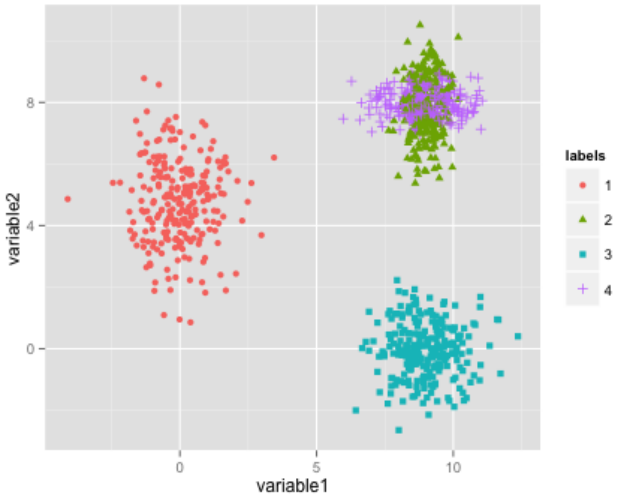

FIGURE 2.1 – Données simulées sous un mélange de quatre lois normales bivariées. Les couleurs et formes des points correspondent aux différentes composantes du modèle de mélange.

réalisations issues de variables aléatoires $\mathbf{Y}^1, \ldots, \mathbf{Y}^p$. On souhaite inférer les dépendances entre les p variables aléatoires à partir des n observations, $\mathbf{y}_1, \ldots, \mathbf{y}_n$. Pour représenter et visualiser ces dépendances, on considère le graphe $G = (V, E)$ où $V = \{1, \ldots, p\}$ est l'ensemble des nœuds représentant les variables aléatoires $\mathbf{Y}^1, \ldots, \mathbf{Y}^p$ et $E \subset V \times V$ est l'ensemble des arêtes du graphe représentant les dépendances entre les variables aléatoires :

$$E = \{(j, j'); j \in V, j' \in V, j \neq j' \text{ tels que les variables } \mathbf{Y}^j \text{ et } \mathbf{Y}^{j'} \text{ sont dépendantes } \}.$$

Il existe plusieurs méthodes d'inférence de réseaux, dont nous exposons brièvement le principe.

Réseaux de dépendance marginale

Ces réseaux de corrélation simples sont également appelés **graphes de covariance** (*covariance graph* ou *relevance network*) (Hastie et al., 2001). Ils modélisent des dépendances marginales en s'intéressant à la distribution jointe de chaque paire de variables. Deux variables aléatoires $\mathbf{Y}^j$ et $\mathbf{Y}^{j'}$ sont indépendantes marginalement si l'on peut écrire la densité jointe des deux variables aléatoires comme le produit de deux densités $p(\mathbf{y}^j; \mathbf{y}^{j'}) = p(\mathbf{y}^j)p(\mathbf{y}^{j'})$ où $p(\mathbf{y}^j; \mathbf{y}^{j'})$, $p(\mathbf{y}^j)$ et $p(\mathbf{y}^{j'})$ sont respectivement les densités des lois de $(\mathbf{y}^j; \mathbf{y}^{j'})$, $\mathbf{y}^j$ et $\mathbf{y}^{j'}$. Les réseaux de dépendance marginales ne sont pas orientés. Dans le cas particulier où les variables aléatoires $\mathbf{Y}^1, \ldots, \mathbf{Y}^p$ suivent des lois normales, le vecteur $\mathbf{Y}_i = (Y_i^1, \ldots, Y_i^p)$ suit une loi normale multivariée $N_p(\mu, \mathbf{\Sigma})$. Dans un contexte de grande dimension, inférer le réseau de dépendance marginale revient à calculer l'estimateur régularisé de la matrice $\mathbf{\Sigma}$. Le problème d'estimation de cette matrice $\mathbf{\Sigma}$ est non convexe mais des méthodes d'estimation ont été proposées (Chaudhuri et al., 2007; Bien and Tibshirani, 2011).

Réseaux basés sur l'information mutuelle

La corrélation simple entre deux variables aléatoires capture uniquement les dé-

pendances linéaires entre des variables. Les réseaux basés sur l'information mutuelle $I(\mathbf{Y}^j, \mathbf{Y}^{j'})$ prennent en compte des dépendances non linéaires.

$$I(\mathbf{Y}^j, \mathbf{Y}^{j'}) = \int\int p(\mathbf{y}^j, \mathbf{y}^{j'}) \log \frac{p(\mathbf{y}^j, \mathbf{y}^{j'})}{p(\mathbf{y}^j)\, p(\mathbf{y}^{j'})}\, d\mathbf{y}^j d\mathbf{y}^{j'}.$$

Ces réseaux ont été étudié par Meyer et al. (2008) et Butte et al. (2000). Dans cette thèse, nous ne nous intéressons pas aux méthodes basées sur l'information mutuelle.

Réseaux bayésiens

Ces modèles sont aussi appelés modèles graphiques orientés. Les arêtes entre les différents nœuds sont orientées, en considérant les ordres de conditionnement entre les variables aléatoires. Par exemple, on considère trois variables aléatoires $\mathbf{Y}^1, \mathbf{Y}^2, \mathbf{Y}^3$ de réalisations $\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3$ et la factorisation suivante des densités de probabilités :

$$p(\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3) = p(\mathbf{y}^1 \mid \mathbf{y}^2, \mathbf{y}^3) p(\mathbf{y}^2 \mid \mathbf{y}^3) p(\mathbf{y}^3)$$

Dans le réseau bayésien correspondant, il existe une arête orientée du nœud 3 vers le nœud 2 due au facteur $p(\mathbf{y}^2 \mid \mathbf{y}^3)$. On dit que le nœud 3 est un parent du nœud 2 et que 2 est un nœud enfant du nœud 3. Les parents du nœud 1 sont $\{2, 3\}$: le réseau comporte donc une arête du nœud 2 vers le nœud 1 et une arête du nœud 3 vers le nœud 1. En revanche, aucune arête n'est dirigée vers le nœud 3. La structure d'un réseau bayésien est un graphe acyclique orienté. Cette structure est induite par la factorisation de la densité sous la forme suivante, où $\text{pa}(\mathbf{y}^j)$ désigne les parents du nœud j :

$$p(\mathbf{y}^1, \ldots, \mathbf{y}^p) = \prod_{j=1}^{p} p(\mathbf{y}^k \mid \text{pa}(\mathbf{y}^j)).$$

Ces réseaux ont été introduits par Pearl (1990) et sont l'objet de nombreux développements méthodologiques (Friedman et al., 2000). De manière générale, les graphes orientés sont plus compliqués à manipuler et à inférer que les graphes non orientés. On peut cependant remarquer que certaines méthodes d'inférence de réseaux non orientés peuvent servir à l'inférence de réseaux orientés (Hastie et al., 2015). Nous ne nous intéressons pas à ce type de réseau dans cette thèse et concentrons notre attention sur les réseaux non orientés.

Réseaux de dépendance conditionnelle

Les modèles graphiques décrivant les relations de dépendance conditionnelle entre des variables aléatoires sont appelés des **champs de Markov** ou des **réseaux de Markov**, ou plus simplement des **modèles graphiques non orientés** (Hastie et al., 2001). Une définition de la notion de dépendance conditionnelle sous-jacente aux modèles graphiques non orientés est détaillée par Bishop (2006). On note $\mathbf{Y}^A = \{\mathbf{Y}^j; j \in A\}$. Le vecteur $\mathbf{Y} = (\mathbf{Y}^1, \ldots, \mathbf{Y}^p)$ vérifie la propriété de Markov par rapport au graphe $G = (V, E)$ s'il vérifie la proposition suivante : pour tout ensemble de nœuds S séparant le graphe en deux sous ensembles de nœuds disjoints A et B, $\mathbf{Y}^A$ et $\mathbf{Y}^B$ sont indépendants conditionnellement à $\mathbf{Y}^S$:

$$p(\mathbf{y}^A; \mathbf{y}^B \mid \mathbf{y}^S) = p(\mathbf{y}^A \mid \mathbf{y}^S) p(\mathbf{y}^B \mid \mathbf{y}^S).$$

Par ailleurs, on définit la factorisation de la densité p du vecteur $(\mathbf{Y}^1, \ldots, \mathbf{Y}^p)$ par rapport au graphe G de la manière suivante. Une clique $\mathcal{C}$ est un ensemble de nœuds du graphe totalement connectés entre eux : pour tout $j, j' \in \mathcal{C}$, $(j, j') \in E$. On note $\mathfrak{C}$ l'ensemble des cliques du graphe G. Pour chaque clique $\mathcal{C} \in \mathfrak{C}$, on définit une fonction de potentiel (ou *potential function*) notée ψ_C et associant à $\mathbf{y}^{\mathcal{C}} = \{\mathbf{y}^j; j \in \mathcal{C}\}$ un nombre réel strictement positif. La distribution p se factorise sur le graphe G si l'on peut écrire :

$$p(\mathbf{y}^1, \ldots, \mathbf{y}^p) = \frac{1}{Z} \prod_{\mathcal{C} \in \mathfrak{C}} \psi_{\mathcal{C}}(\mathbf{y}^{\mathcal{C}}), \tag{2.4}$$

où la quantité Z est appelée fonction de partition. Il s'agit d'un facteur de normalisation garantissant que la fonction p soit bien une densité de probabilité. Dans cette expression, les fonctions ψ_C ne sont pas des densités de probabilité.
Le théorème de Hammersley-Clifford montre l'équivalence entre cette factorisation de la densité p du vecteur aléatoire $\mathbf{Y}$ par rapport au graphe G et la propriété de Markov du vecteur aléatoire $\mathbf{Y}$ par rapport au graphe G.
La notion d'indépendance conditionnelle entre les variables aléatoires est ainsi directement liée à la structure du graphe. Les modèles graphiques non orientés sont plus informatifs que les réseaux de dépendance marginale présentés précédemment. Nous concentrons notre attention sur ces modèles graphiques non orientés. On distingue les modèles graphiques en fonction de la nature, discrète ou continue, des variables aléatoires considérées : modèle graphique gaussien ou modèle d'Ising.

2.2.2 Les modèles graphiques non-orientés

Le modèle graphique gaussien

Dans le cadre du modèle graphique gaussien, les données sont un échantillon de taille n issu d'une loi normale mutivariée de dimension p, de moyenne nulle $\mathbf{0}$ et de variance $\mathbf{\Sigma}$, matrice définie positive de taille $p \times p$. Les observations $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$ sont indépendantes et identiquement distribuées :

$$\mathbf{y}_i \sim \mathcal{N}_p(\mathbf{0}, \mathbf{\Sigma}) \text{ pour tout } i \in 1, \ldots, n.$$

Dans ce modèle, les dépendances conditionnelles entre deux variables $\mathbf{Y}^j$ et $\mathbf{Y}^{j'}$ conditionnellement aux autres variables sont directement liées aux coefficients de la matrice la matrice de covariance inverse, notée $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$. En effet, le coefficient de corrélation partielle $\rho_{jj'}$ entre les variables j et j' vérifie la relation suivante : $\rho_{jj'} = \frac{\theta_{jj'}}{\sqrt{\theta_{jj}\theta_{j'j'}}}$. Un coefficient nul $\theta_{jj'} = 0$ indique l'indépendance des variables j et j' conditionnellement à toutes les autres variables du jeu de données (Whittaker, 1990; Lauritzen, 1996). Inférer le graphe de dépendance entre les p variables revient donc à détecter les coefficients non nuls de la matrice $\mathbf{\Theta}$. Ce problème est connu sous le nom de sélection de covariance, en anglais *covariance selection model* (Dempster, 1972) ou *Gaussian concentration graph model*. La matrice $\mathbf{\Theta}$ est aussi appelée matrice de concentration.

Notons que l'on peut présenter cette distribution gaussienne multivariée sous la forme d'un modèle graphique à l'aide de la factorisation de la densité suivante, comme présentée à la formule (2.4) :

$$p(\mathbf{y}^1, \ldots, \mathbf{y}^p) = \exp\left(-\frac{1}{2}\sum_{j,j' \in V} \theta_{jj'}\mathbf{y}^j\mathbf{y}^{j'} - A(\boldsymbol{\Theta})\right),$$

où $A(\boldsymbol{\Theta}) = -\frac{1}{2}\log\det\left[\frac{\boldsymbol{\Theta}}{2\pi}\right]$. Cette fonction $A(\boldsymbol{\Theta})$ correspond au logarithme de la fonction de partition $Z(\Theta)$. Cette écriture permet de voir que la factorisation du vecteur aléatoire $\mathbf{Y}$ par rapport au graphe G est déterminée par les paramètres $\theta_{jj'}$. La structure du graphe à inférer est donc totalement déterminée par $\boldsymbol{\Theta}$.

Une première approche pour obtenir la structure du graphe consiste à calculer $\boldsymbol{S}$ la matrice de variance covariance empirique et d'inverser cette matrice afin d'obtenir une estimation de la matrice $\boldsymbol{\Theta}$:

$$\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^T,$$

où $\bar{\mathbf{y}}$ est le vecteur de moyenne empirique. Lorsque le nombre d'échantillons est inférieur au nombre de variables, cette approche n'est pas possible puisque la matrice $\boldsymbol{S}$ n'est pas inversible. Une autre approche consiste à tester l'inclusion ou l'exclusion des arêtes du graphe à l'aide d'un algorithme de type forward ou backward (Dempster, 1972) mais cette approche est coûteuse et peu efficace. En grande dimension (cas $n < p$), on distingue deux grands types de méthodes d'inférence du graphe : certaines méthodes sont basées sur des tests multiples (Schäfer and Strimmer, 2005; Wille and Peter, 2006) tandis que d'autres méthodes sont basées sur des méthodes de régularisation ℓ_1 (Meinshausen and Buhlmann, 2006; Huang et al., 2006; Banerjee et al., 2008; Friedman et al., 2008). Nous concentrons notre attention sur ces méthodes d'inférence de graphe à l'aide de pénalisation ℓ_1, en détaillant d'abord le lasso graphique (ou *graphical lasso*) basé sur le calcul de la vraisemblance pénalisée exacte, puis la sélection au voisinage d'un nœud (ou *neighbourhood selection*) basée sur le calcul de régressions pénalisées.

Le lasso graphique

On considère la log-vraisemblance pénalisée suivante, où λ est le paramètre de régularisation de la pénalité ℓ_1 imposée sur les éléments de la matrice $\boldsymbol{\Theta}$ à estimer.

$$L_\lambda(\boldsymbol{\Theta}) = \log\det(\boldsymbol{\Theta}) - \text{trace}(\boldsymbol{S\Theta}) - \lambda \mid\mid \boldsymbol{\Theta} \mid\mid_1 . \tag{2.5}$$

Le problème d'estimation de la matrice $\boldsymbol{\Theta}$ à partir de cette vraisemblance pénalisée est connu sous le nom du lasso graphique ou *graphical lasso*. Il s'agit d'un problème d'estimation convexe. Pour le résoudre de manière efficace sur des problèmes de grande dimension, Banerjee et al. (2008) et Friedman et al. (2008) ont proposé un algorithme de descente du gradient par bloc. Pour le choix du paramètre de régularisation λ, Banerjee et al. (2008) ont proposé un critère de sélection de modèle. Pour un seuil fixé $\alpha \in]0, 1[$, le choix de paramètre de régularisation $\lambda(\alpha)$ garantit que la probabilité de joindre par erreur deux composantes connexes disjointes du réseau soit bornée par α :

$$\lambda(\alpha) = (\max_{j'>j}\hat{\sigma}_{j'}\hat{\sigma}_j)\frac{t_{n-2}(\alpha/2p^2)}{\sqrt{n-2+t^2_{n-2}(\alpha/2p^2)}},$$

où $t_{n-2}(\alpha)$ désigne le quantile d'ordre $(100-\alpha)$ de la distribution de student à $n-2$ degrés de liberté, et $\hat{\sigma}_j$ est la variance empirique de la variable j.

D'autres choix de paramètres de régularisation ont été proposés (*Stability Approach to Regularization Selection criterion* (StARS), extendedBIC ...) et les développements méthodologiques autour de ce choix sont nombreux.

Sélection de voisinage (ou *neighbourhood selection*)

Une méthode alternative au lasso graphique est la sélection d'arêtes au voisinage de chaque nœud du graphe à l'aide de régression pénalisée. Meinshausen and Buhlmann (2006) ont utilisé le lien existant entre les coefficients dans la régression de la variable j sur les autres variables du réseau et les coefficients de la matrice $\mathbf{\Theta}$:

$$\mathbf{y}^j = \sum_{j' \neq j} \beta_{jj'} \mathbf{y}^{j'} + \epsilon_j,$$

$$\beta_k^j = \frac{\theta_{jk}}{\theta_{jj}}.$$

Les paramètres $\beta_{jj'}$ sont les coefficients dans la régression de $\mathbf{y}^j$ sur les autres variables et $\epsilon_j \sim \mathcal{N}(0, \sigma_j)$. Dans le contexte de grande dimension, Meinshausen and Buhlmann (2006) proposent d'effectuer des régressions pénalisées de chaque variable du réseau sur les autres variables afin d'identifier les coefficients $\theta_{jj'}$ non nuls. Les coefficients $\theta_{jj'}$ sont estimées non nuls si au moins un des coefficients $\beta_{jj'}$ et $\beta_{j'j}$ est non nul (*OR rule*), ou, de manière alternative, si ces deux coefficients sont non nuls (*AND rule*).

Pour le choix du paramètre de régularisation λ dans chaque régression pénalisée, le paramètre $\lambda(\alpha)$, pour $\alpha \in]0,1[$, garantit que la probabilité de joindre par erreur deux composantes connexes disjointes du réseau soit borné par α :

$$\lambda(\alpha) = \frac{2\hat{\sigma}_j}{\sqrt{n}} \tilde{\Phi}^{-1}\left(\frac{\alpha}{2pn^2}\right),$$

où $\tilde{\Phi} = 1 - \Phi$, Φ est la distribution cumulée de la la normale $\mathcal{N}(0,1)$ et $\hat{\sigma}_j$ est la variance empirique de la variable j. Cette procédure est asymptotiquement consistante pour l'estimation des éléments non nuls de la matrice $\mathbf{\Theta}$ (Meinshausen and Buhlmann, 2006).

Les modèles graphiques discrets

On considère un ensemble de variables aléatoires $\mathbf{Y}^1, \ldots, \mathbf{Y}^p$ prenant ses valeurs dans un ensemble discret $\{0,1\}$. Ce modèle graphique discret, aussi appelé modèle d'Ising, est un exemple de modèle graphique discret. La factorisation de la densité suivante par rapport au graphe G s'écrit de la manière suivante :

$$p(\mathbf{y}^1, \ldots, \mathbf{y}^p) = \exp\left(\sum_{j \in V} \theta_{jj} \mathbf{y}^j + \sum_{(j,j') \in E} \theta_{jj'} \mathbf{y}^j \mathbf{y}^{j'} - A(\mathbf{\Theta})\right),$$

$$A(\mathbf{\Theta}) = \log \left[\sum_{y \in \{0,1\}^p} \exp \left(\sum_{j \in V} \theta_{jj} \mathbf{y}^j + \sum_{(j,j') \in E} \theta_{jj'} \mathbf{y}^j \mathbf{y}^{j'} \right) \right].$$

Ce modèle graphique discret est aussi appelé modèle d'Ising. Contrairement au modèle graphique gaussien, pour lequel la fonction de partition $A(\mathbf{\Theta})$ se calcule bien, le calcul de $A(\mathbf{\Theta})$ dans le cas discret implique une sommation sur 2^p termes. Une estimation des paramètres $\mathbf{\Theta}$ similaire à celle du lasso graphique est donc impossible. La méthode d'inférence de réseaux basée sur la sélection de voisinage est donc particulièrement intéressante dans le cas discret. Pour le modèle d'Ising, cette estimation revient à effectuer, pour chaque variable $\mathbf{y}^j$ la régression logistique suivante, en ajoutant une pénalité ℓ_1 pour forcer la nullité de certains coefficients $\mathbf{\Theta}$:

$$\log \left[\frac{p(\mathbf{y}^j = 1 \mid \mathbf{y}^{V \setminus j})}{p(\mathbf{y}^j = 0 \mid \mathbf{y}^{V \setminus j})} \right] = \theta_{jj} + \sum_{j' \in V, j' \neq j} \theta_{jj'} \mathbf{y}^{j'}.$$

Chapitre 3

Contributions

Notations

Dans cette thèse, l'indexation des données $\mathbf{y}$ dépend du modèle statistique considéré, en conservant les conventions suivantes : l'indice $i = 1, \ldots, n$ indexe toujours les unités statistiques ou observations, l'indice $j = 1, \ldots, p$ indexe toujours les variables. En fonction du type de modèle considéré, l'indice j indexe les gènes (analyse différentielle, inférence de réseau) ou les échantillons (classification par modèle de mélange). Le tableau 3.1 résume les changements d'indices des gènes et échantillons en fonction du modèle considéré.

	Unités statistiques ou observations $i \in 1, \ldots, n$	Variables $j \in 1, \ldots, p$
Analyse différentielle	échantillons	gènes
Classification	gènes	échantillons
Inférence de réseaux	échantillons	gènes

Table 3.1 – Indices des gènes et échantillons en fonction du type d'analyse effectuées

3.1 Filtrage des données RNA-seq

On rappelle que l'on dispose d'une matrice $\mathbf{y}$ de taille $n \times p$ correspondant à n échantillons $\mathbf{y}_i$ pour $i = 1, \ldots, n$ pour p gènes $\mathbf{y}^j$ pour $j = 1, \ldots, p$. La quantité y_{ij} représente l'expression du gène j pour l'échantillon i. Dans le cadre de l'analyse différentielle des données d'expression de gènes (section 1.2.2), l'étape de correction des p-values pour les tests multiples dans la détection de gènes différentiellement exprimés est une étape importante afin de garder un taux de détection de faux positifs relativement bas. Cependant, ce contrôle du nombre de faux positifs se fait au détriment de la capacité du test à détecter les gènes réellement différentiellement exprimés, aussi appelée puissance de détection. Pour cette raison, il est essentiel de retirer, avant toute analyse différentielle, les gènes dont le profil d'expression ne parait pas informatif. On considère l'exemple suivant (tableau 3.2) : le gène 4 comporte un alignement de 48 *reads* pour l'échantillon 3 de la condition 1, et uniquement pour cet échantillon. Ce gène ne semble pas apporter d'information et pourra difficilement être considéré comme différentiellement exprimé entre la condition 1 et la condition 2. Il serait donc raisonnable de supprimer le gène 4 avant d'effectuer l'analyse différentielle.

		gène 1	gène 2	gène 3	gène 4	gène 5	gène 6	...
condition 1	échantillon 1	4	199	2987	0	65	1905	...
	échantillon 2	0	189	1806	0	29	121	...
	échantillon 3	6	201	1752	48	599	169	...
	échantillon 4	4	198	2987	0	65	1905	...
condition 2	échantillon 1	0	0	1296	0	49	121	...
	échantillon 2	6	0	2298	0	119	169	...
	échantillon 3	4	0	2987	0	651	1905	...
	échantillon 4	0	0	1876	0	219	121	...

TABLE 3.2 – Tableau de comptages résumant l'abondance des *reads* par région génomique d'intérêt pour deux conditions expérimentales avec quatre réplicats biologiques par condition.

Les méthodes de filtrage utilisées en pratique consistent à supprimer les gènes :
- dont la somme totale des *reads* est inférieure à un certain seuil fixé arbitrairement (Sultan et al., 2008)
- ayant au moins un comptage nul dans chaque condition expérimentale (Bottomly et al., 2011),
- dont la valeur moyenne exprimée en RPKM est inférieure à une certaine valeur fixée arbitrairement (Mortazavi et al., 2008),
- dont la valeur maximale exprimée en CPM est inférieure à une certaine valeur fixée arbitrairement (Robinson and Oshlack, 2010) .

Toutes ces méthodes sont basées sur le choix d'un seuil à fixer arbitrairement. Nous avons proposé un niveau de seuil des données qui ne soit pas arbitrairement choisi. Ce seuil est ajusté aux données et correspond à la valeur maximisant un indice de similarité entre les réplicats biologiques d'une même condition. On considère deux réplicats biologiques i et i' appartenant à une même condition expérimentale, un seuil fixé s et le tableau de contingence tableau 3.3. Pour un seuil donné, on calcule un indice de Jaccard, coefficient de similarité entre les deux réplicats i et i' à partir du tableau de contingence

tableau 3.3 :

$$J_s(\mathbf{y}_i, \mathbf{y}_{i'}) = \frac{a}{a+b+c}.$$

		Réplicat i	
		Nombre de gènes j tels que $y_{ij} > s$	Nombre de gènes j tels que $y_{ij} \leq s$
Réplicat i'	Nombre de gènes j tels que $y_{i'j} > s$	a	b
	Nombre de gènes j tels que $y_{i'j} \leq s$	c	d

TABLE 3.3 – Tableau de contingence du nombre de gènes dont le comptage normalisé est inférieur/ supérieur au seuil s pour deux échantillons i et i'.

On note T le cardinal de l'ensemble des couples d'échantillons possibles appartenant à une même condition. On calcule la moyenne de ces coefficients sur l'ensemble des couples d'échantillons appartenant à une même condition possible :

$$J_s^\star(\mathbf{y}) = \frac{1}{T} \sum_{\substack{i<i' \\ c(i)=c(i')}} J_s(\mathbf{y}_i, \mathbf{y}_{i'}).$$

Le seuil à choisir, guidé par les données, est celui qui maximise la quantité $J_s^\star(\mathbf{y})$:

$$s^\star = \underset{s}{\operatorname{argmax}}\, J_s^\star(\mathbf{y}).$$

Le choix du seuil selon cette méthode garantit que la similarité entre les profils d'expression des échantillons d'une même condition expérimentale soit la plus forte possible. Le seuil fixé est spécifique à chaque jeu de données. Les caractéristiques statistiques et pratiques de ce seuil sont détaillés dans le chapitre 4 qui correspond à l'article publié suivant (Rau et al., 2013).

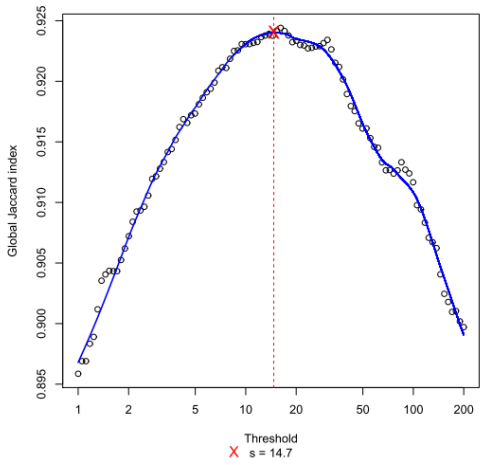

FIGURE 3.1 – Indice global $J_s^\star(\mathbf{y})$ calculé sur les données normalisées Bottomly (Bottomly et al., 2011) pour un seuil variant de 1 à 200. La courbe bleue est une courbe LOESS (*locally weighted scatterplot smoothing*). La croix rouge indique le maximum de l'indice $J_s^\star(\mathbf{y})$ et la ligne rouge indique le seuil correspondant et sélectionné. Cette figure est extraite de Rau et al. (2013).

3.2 Classification des données d'expression par modèle de mélange

On rappelle que l'on dispose d'une matrice de mesures d'expression de n gènes $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$. Pour chaque gène $i = 1, \ldots, n$, le vecteur $\mathbf{y}_i$ indique l'expression du gène i pour les p conditions expérimentales $j = 1, \ldots, p$. On souhaite classer les gènes en fonction de leur niveau d'expression à travers les différentes conditions expérimentales à l'aide d'un modèle de mélange. Dans ce contexte, les unités statistiques sont les gènes $\mathbf{y}_1, \ldots, \mathbf{y}_n$ tandis que les échantillons $\mathbf{y}^1, \ldots, \mathbf{y}^p$ correspondent aux variables du modèle (voir tableau 3.1). Contrairement à l'analyse différentielle ou à l'inférence de réseaux, le faible nombre d'échantillons disponibles (p petit) ne pose pas de problème statistique puisque nous classons les gènes et non les échantillons.

3.2.1 Transformation des données et comparaison de modèles pour la classification des données RNA-seq

Comme détaillé en section 2.1.2, on suppose que les données $\mathbf{y}$ sont la réalisation d'un mélange de K variables aléatoires :

$$f(\mathbf{y}_i; K, \theta_K) = \sum_{k=1}^{K} p_k f_k(\mathbf{y}_i; \mathbf{a}_k).$$

Le choix de la densité f_k est une question importante. Le tableau 3.4 récapitule les différentes lois et méthodes utilisées sur les données de puces à ADN et les données RNA-seq. Pour l'analyse différentielle, Robinson and Oshlack (2010) et Anders and Huber (2010) ont d'abord proposé des modèles basés sur des lois discrètes, puis Law et al. (2014) ont proposé une transformation des données RNA-seq et l'utilisation du modèle basé sur des lois gaussiennes initialement développé pour l'analyse différentielle des données de puces à ADN. Le choix entre l'utilisation de modèle discret (`edgeR`, `DESeq2`) ou continu (`lima` + `"voom"`) dépend principalement du design de l'expérience et du nombre de réplicats disponibles.

Pour l'analyse de co-expression des données RNA-seq, Rau et al. (2015) ont proposé un modèle de mélange de lois de Poisson. Nous proposons une transformation simple des données RNA-seq permettant l'utilisation du modèle de mélange gaussien, modèle bien établi pour l'analyse des données de puces à ADN (Yeung et al., 2001). Afin de choisir entre l'utilisation du modèle gaussien sur données transformées et l'utilisation du modèle de Poisson, nous proposons de calculer le critère BIC de chaque modèle et de comparer ces BIC. Le calcul du critère BIC est effectué en prenant en compte la transformation des données. La prise en compte de la transformation des données dans le calcul du BIC a déjà été proposé par Thomas et al. (2008) en analyse de données d'urbanisme. La prise en compte de la transformation dans le calcul a un intérêt très pratique pour les données RNA-seq. Nous illustrons l'utilité de cette comparaison de modélisation discrète ou continue dans le chapitre 5. Ce chapitre a fait l'objet d'une communication orale aux 47èmes Journées de Statistique de la SFdS.

	Données de puces *données continues* 	Données RNA-seq *données de comptage* 
ANALYSE DIFFÉRENTIELLE	Gaussienne : `limma` *(Smyth & al, 2005)*	Binomiale Négative : `DESeq` ; `EdgeR` *(Anders & al, 2010 ; Robinson & al, 2010)*, ou Gaussienne *sur données transformées* : `limma` + "voom" *(Law & al, 2014)*
CLASSIFICATION PAR MODÈLE DE MÉLANGE	Gaussienne : `Rmixmod` *(Yeung & al, 2000, Biernacki & al, 2006)*	Poisson : `HTSCluster` *(Rau & al, 2015)* ou Gaussienne *sur données transformées ?*

TABLE 3.4 – Tableau récapitulatif des lois utilisées et des packages R implémentant les méthodes correspondantes pour l'analyse différentielle et la classification par modèle de mélange.

3.2.2 Un critère de sélection de modèle pour la classification par modèle de mélange de données annotées d'expression de gènes.

En classification des données RNA-seq par modèle de mélange, le problème statistique est inversé par rapport aux problèmes d'analyse différentielle ou d'inférence de réseaux : les observations à classer $\mathbf{y}_1, \ldots, \mathbf{y}_n$ sont les gènes et les variables $\mathbf{y}^1, \ldots, \mathbf{y}^p$ sont les échantillons. Dans ce contexte, le manque d'échantillons, caractéristique des jeux de données RNA-seq à notre disposition, ne pose donc pas de problème statistique. Ce faible nombre d'échantillons (*i.e.* de variables) nous incite cependant à chercher des variables informatives externes sur la nature des gènes afin d'améliorer la qualité de la classification. Ces variables informatives externes peuvent être extraites des bases de données d'annotations de gènes, décrites en section 1.2.3. Une première approche consiste à inclure ces données directement dans le modèle en les considérant comme des variables au même titre que les échantillons. Une autre approche consiste à utiliser ces annotations pour améliorer l'estimation des paramètres du modèle de mélange (Pan, 2006; Huang et al., 2006). Cependant, ces annotations sont généralement utilisées pour valider *a posteriori* la pertinence des modules détectés. Nous préférons donc utiliser ces variables externes uniquement dans l'étape de sélection de modèle, et non dans l'étape d'estimation des paramètres du modèle de mélange. Une première solution pour intégrer ces annotations externes dans l'étape de sélection de modèle est l'utilisation du critère SICL. Nous détaillons le principe du critère SICL ci-dessous.

Le critère SICL

Baudry et al. (2014) a proposé un critère de type ICL prenant en compte plusieurs variables catégorielles externes pouvant éventuellement contribuer à expliquer la partition des données inférée. On note ces variables externes $\mathbf{u} = (\mathbf{u}^1, \ldots, \mathbf{u}^R)$ avec $r = 1, \ldots, R$

où $u_{i\ell}^r = 1$ indique que l'observation i appartient à la catégorie ℓ de la $r^{\text{ième}}$ variable externe catégorielle, $u_{i\ell}^r = 0$ sinon. On souhaite obtenir une classification $\mathbf{z}$ à partir des données $\mathbf{y}$ telle que la classification $\mathbf{z}$ soit la plus cohérente possible avec les variables catégorielles externes $\mathbf{u}$. En supposant que $\mathbf{y}$ et $\mathbf{u}$ sont conditionnellement indépendants sachant la classification $\mathbf{z}$, le critère de classification supervisée *Supervised Integrated Completed Likelihood* (SICL) est une approximation asymptotique du logarithme de la log-vraisemblance complétée :

$$f(\mathbf{y}, \mathbf{u}, \mathbf{z}; K) = \int f(\mathbf{y}, \mathbf{u}, \mathbf{z}; K, \theta_K)\pi(\theta_K)d\theta_K.$$

Le critère SICL est défini de la manière suivante :

$$\text{SICL}(K; \mathbf{y}) = \text{ICL}(K) + \sum_{r=1}^{R}\sum_{\ell=1}^{U_r}\sum_{k=1}^{K} n_{k\ell}^r \log\frac{n_{k\ell}^r}{n_{k\cdot}}, \tag{3.1}$$

où U_r est le nombre de catégories de la variable catégorielle externe $\mathbf{u}^r$,

$$n_{k\ell}^r = \text{card}\big\{i : z_{ik} = 1 \text{ and } u_{i\ell}^r = 1\big\},$$

et $n_{k\cdot} = \sum_{\ell=1}^{U_r} n_{k\ell}^r$. Le terme additional dans l'Equation (3.1) mesure la force du lien entre les variables catégorielles externes $\mathbf{u}$ et la classification $\mathbf{z}$.

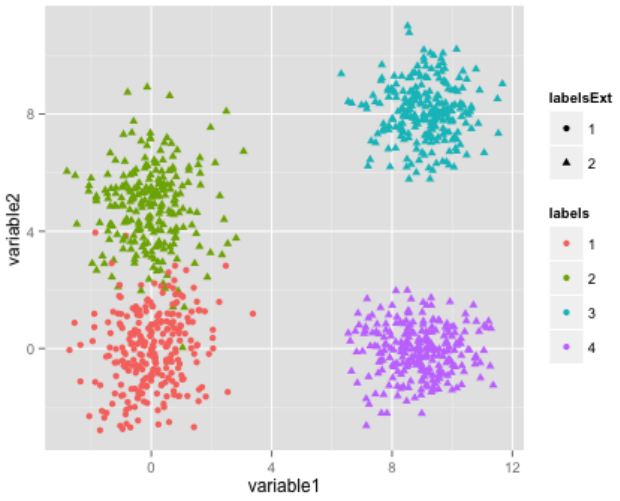

FIGURE 3.2 – Données simulées sous un mélange de 4 lois normales bivariées. Les couleurs des points correspondent aux différentes composantes du modèle de mélange tandis que les formes indiquent l'appartenance aux catégories d'une classification externe en deux catégories.

Les données représentées en Figure 3.2 sont simulées à partir d'une mélange de 4 lois gaussiennes bivariées représentées par les 4 différentes couleurs. Par ailleurs, on dispose d'une variable catégorielle externe indiquant l'appartenance des points à une classification externe. Sans prendre en compte cette information externe, le critère ICL sélectionne une partition à trois classes, les deux composantes à gauche de la figure sont

rassemblées une seule classe. Le critère SICL, prenant en compte cette information externe, sélectionne une classification à 4 classes, isolant la composante en bas à gauche de la figure (cercles) des points en haut à gauche de la classification (triangles). La partition sélectionnée par le critère SICL est ainsi plus facilement interprétable en fonction de la classification externe symbolisée par les formes des points.

Le critère ICAL

Même si les termes d'annotations sont des variables binaires (*i.e.* un gène est annoté ou n'est pas annoté), les bases de données d'annotation fonctionnelle sont souvent incomplètes ce qui rend l'utilisation du critère SICL peu approprié. On considère un ensemble de G termes d'annotation extrait de ces bases de données d'annotation. Ces termes sont indexés par l'indice g, l'information extraite des bases de données pour chaque terme g est notée $\mathbf{u}^g$ où :

$$u_i^g = \begin{cases} 1 & \text{si le gène } i \text{ est référencé pour l'annotation fonctionnelle } g, \\ 0 & \text{si le gène } i \text{ n'est pas référencé pour l'annotation fonctionnelle } g. \end{cases}$$

Les annotations ne peuvent pas être considérées comme des variables catégorielles binaires (présence/absence d'annotations) car une absence d'annotation peut signifier deux choses : soit le gène n'est pas annoté pour la fonction car il ne la possède pas, soit le gène n'a pas encore été identifié comme annoté. Le critère de sélection de modèle SICL, qui prend en compte des variables catégorielles externes, n'est donc pas pertinent. C'est pour cette raison que nous avons proposé le critère *Integrated Completed Annotated Likelihood* (ICAL). Pour cela, on définit, pour chaque terme d'annotation g, la matrice aléatoire $\mathbf{b}^g$, variable latente indiquant la répartition des gènes annotés dans les K classes :

$$b_{ik}^g = \begin{cases} 1 & \text{avec probabilité } p_k^g \text{ if } u_i^g = 1, \\ 0 & \text{si } u_i^g = 0. \end{cases}$$

Le critère ICAL est une approximation de la log-vraisemblance complétée annotée des données :

$$\log f(\mathbf{y}, \mathbf{z}, \mathbf{b}^1, \ldots, \mathbf{b}^G; K) = \log \int_{\theta_K} f(\mathbf{y}, \mathbf{z}, \mathbf{b}^1, \ldots, \mathbf{b}^G; K, \theta_K) \pi(\theta_K) d\theta_K.$$

On définit $n^g = \text{card}\{i : u_i^g = 1\}$ et $n_k^g = \text{card}\{i : \hat{z}_{ik} = 1 \text{ et } u_i^g = 1\}$. Ce critère s'écrit :

$$\text{ICAL}(K; \mathbf{y}) = \text{ICL}(K) + \sum_{g=1}^{G} \sum_{k=1}^{K} n_k^g \log \frac{n_k^g}{n^g}.$$

Ce critère est présenté au chapitre 6 et fait l'objet d'une publication acceptée (Gallopin et al., 2015).

3.3 Inférence de réseaux à l'aide de modèle graphique

On rappelle que l'on dispose d'une matrice $\mathbf{y}$ de taille $n \times p$ correspondant à n échantillons $\mathbf{y}_i$ pour $i = 1, \ldots, n$ pour p gènes $\mathbf{y}^j$ pour $j = 1, \ldots, p$. On suppose que les vecteurs $\mathbf{y}^1, \ldots, \mathbf{y}^p$ sont n réalisations issus de variables aléatoires $\mathbf{Y}^1, \ldots, \mathbf{Y}^p$ représentant le niveau d'expression des gènes. On souhaite inférer les dépendances entre les p variables aléatoires à partir des n échantillons $\mathbf{y}_1, \ldots, \mathbf{y}_n$. Pour représenter et visualiser ces dépendances, on considère le graphe $G = (V, E)$ où $V = \{1, \ldots, p\}$ est l'ensemble des nœuds représentant les variables aléatoires $\mathbf{Y}^1, \ldots, \mathbf{Y}^p$ et $E \subset V \times V$ est l'ensemble des arêtes du graphe représentant les dépendances entre les variables aléatoires :

$$E = \{(j, j'); j \in V, j' \in V, j \neq j' \text{ tels que les variables } \mathbf{Y}^j \text{ et } \mathbf{Y}^{j'} \text{ sont dépendantes } \}.$$

3.3.1 Un modèle pour l'inférence de réseaux RNA-seq

L'utilisation du modèle graphique gaussien est bien établi en inférence de réseaux à partir de données de puces à ADN. Comme pour l'analyse différentielle et la classification, les modèles d'inférence de réseaux doivent être adaptés au caractère discret des données RNA-seq. Dans le cas du modèle graphique de Poisson, la factorisation de la distribution du vecteur aléatoire $(\mathbf{Y}^1, \ldots, \mathbf{Y}^p)$ par rapport au graphe G est la suivante :

$$p(\mathbf{y}^1, \ldots, \mathbf{y}^p) = \exp\left[\sum_{j \in V}(\beta_j \mathbf{y}^j - \log\left(\mathbf{y}^j!\right)) + \sum_{(j,j') \in E} \beta_{jj'} \mathbf{y}^j \mathbf{y}^{j'} - A(\boldsymbol{\beta})\right]$$

où $A(\boldsymbol{\beta})$ est le logarithme de la fonction de partition $Z(\boldsymbol{\beta})$ garantissant que p soit une distribution (voir (2.4)). Comme dans le cas du modèle d'Ising, le calcul de $A(\boldsymbol{\beta})$ est impossible ne permettant pas l'estimation des paramètres $\boldsymbol{\beta}$ à partir de la vraisemblance pénalisée exacte, comme détaillé en section 2.2.2.

Une approche alternative est la sélection des arêtes aux voisinage de chaque noeud du réseau (*neighbourhood selection*). Allen and Liu (2013) ont proposé l'utilisation de régression linéaire généralisée de Poisson. Ce modèle prend en compte le caractère discret des données RNA-seq, mais il ne prend pas en compte la grande dispersion inter-échantillons. Afin de prendre en compte cette dispersion, nous avons proposé de remplacer la régression linéaire généralisée de Poisson par une régression linéaire généralisée mixte, basée sur une loi de Poisson hiérarchique $\mathbf{Y}^j \sim \mathcal{P}(\boldsymbol{m_j})$ où le paramètre de moyenne $\boldsymbol{m_j}$ est également une variable aléatoire : $\boldsymbol{m_j} = \mu_j \exp(\boldsymbol{\varepsilon}_j)$ avec $\boldsymbol{\varepsilon}_j \sim \boldsymbol{N}_n(0, \sigma_j^2 I_n)$ et μ_j est un vecteur de taille n dont chaque entrée est $\exp(\sum_{j' \neq j} \beta_{jj'} y_{ij'})$ pour $i \in 1, \ldots, n$. Contrairement au modèle graphique gaussien ou au modèle graphique log-linéaire de Poisson, le modèle hiérarchique log-linéaire de Poisson proposé ne requiert pas de transformation préalable des données pour son utilisation sur les données RNA-seq.

Ce travail est exposé dans le chapitre 7 et a fait l'objet d'une publication (Gallopin et al., 2013).

3.3.2 Sélection de covariance diagonale par bloc pour le modèle graphique gaussien en grande dimension.

Le faible nombre d'échantillons disponibles pour les jeux de données RNA-seq limite fortement la performance de tout type de méthode d'inférence de réseaux (modèle graphique gaussien ou modèles graphiques discrets). Ce constat correspond à un résultat déjà connu pour le cas de la régression linéaire gaussienne. Dans ce cadre, Verzelen (2012) a défini les cas d'*ultra-grande dimension* de la manière suivante, où p est le nombre de gènes, n le nombre d'échantillons et d le degré maximal du réseau, *i.e.* le nombre maximal de connexions entre un gène et les autres gènes du réseau :

$$\frac{d \log(\frac{p}{d})}{n} \geq \frac{1}{2}.$$

Dans le contexte d'ultra-grande dimension, il convient donc de réduire le plus possible le nombre de variables à inclure dans le réseau afin de réduire le nombre de paramètres à estimer. Ce constat a été à l'origine de la contribution suivante.

Dans le cadre du spécifique du modèle graphique gaussien, on estime la matrice $\mathbf{\Theta}$ à l'aide du *graphical lasso* (Friedman et al., 2008), afin d'inférer la structure du graphe de dépendances conditionnelles correspondant, comme détaillé en section 2.2.2 :

$$\begin{aligned} \ell_\lambda(\mathbf{\Theta}) &= \log\det(\mathbf{\Theta}) - \mathrm{trace}(S\mathbf{\Theta}) - \lambda \,||\, \mathbf{\Theta} \,||_1 \\ \hat{\mathbf{\Theta}}^{(\lambda)} &= \underset{\mathbf{\Theta}}{\mathrm{argmin}} \left\{ \ell_\lambda(\mathbf{\Theta}) \right\}. \end{aligned} \tag{3.2}$$

Afin de réduire le nombre de paramètres à estimer, Mazumder and Hastie (2012) et Witten et al. (2011) ont remarqué la propriété suivante : si la solution $\hat{\mathbf{\Theta}}^{(\lambda)}$ du problème 3.2 est diagonale par bloc à une permutation des variables près, alors le problème d'inférence peut être résolu dans chaque bloc indépendamment. De plus, il existe un moyen simple de savoir si la solution $\hat{\mathbf{\Theta}}^{(\lambda)}$ a une structure diagonale par bloc : $\hat{\mathbf{\Theta}}^{(\lambda)}$ et la matrice de covariance empirique seuillée $S^{(\lambda)} = (\hat{\sigma}_{jj'} \mathbf{1}_{\{|\hat{\sigma}_{jj'}|>\lambda\}})_{jj'}$ ont la même structure diagonale par bloc. Ce constat est à l'origine de la règle du *screening* diagonal par bloc (*block diagonal screening rule*) pour une valeur fixée du paramètre de régularisation λ :

Etape 1 Identification des blocs, correspondant aux composantes connexes du graphe inféré à partir de la matrice d'adjacence $(\mathbf{1}_{\{|\hat{\sigma}_{jj'}|>\lambda\}})_{jj'}$,

Etape 2 Estimation des paramètres $\hat{\mathbf{\Theta}}_1^{(\lambda)}, \ldots, \hat{\mathbf{\Theta}}_K^{(\lambda)}$ dans chacun des blocs détectés à l'étape 1.

Le paramètre de régularisation λ utilisé à l'étape 1 est le même que les paramètres de régularisation utilisés pour l'estimation des paramètres dans chaque bloc à l'étape 2. Afin d'améliorer les estimations, Tan et al. (2015) ont proposé le *Cluster Graphical Lasso* : à l'étape 1, ils identifient K composantes connexes à l'aide d'un clustering hiérarchique réalisé à partir de la matrice de similarité $\tilde{S} =| S |$. Puis ils infèrent les réseaux dans chaque bloc en utilisant différents paramètres de régularisation $\rho_1, \ldots, \rho_K$. Ils montrent ainsi que l'utilisation de différents paramètres de régularisation λ à l'étape 1 et dans les sous-problèmes de l'étape 2 permet d'améliorer les performances de l'inférence.

Dans cet esprit, nous proposons d'inférer le réseau en deux étapes en utilisant des paramètres de régularisation spécifiques à chaque étape. Nous concentrons notre attention

sur le choix du paramètre de régularisation dans l'étape 1. Nous notons $\mathcal{B}$ l'ensemble des partitions possibles des p variables en différents blocs. Nous reformulons le choix du seuil à appliquer à la matrice de covariance S en un problème de sélection de modèle parmi la collection de modèle suivante $\mathcal{F} = (F_B)_{B\in\mathcal{B}}$:

$$F_B = \left\{ \phi(0,\mathbf{\Sigma}_B) \text{ où } \mathbf{\Sigma}_B = \begin{pmatrix} \mathbf{\Sigma}^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{\Sigma}^K \end{pmatrix}, \mathbf{\Sigma}^k \text{ matrice de taille } p_k \times p_k \text{ pour } k\in\{1,\dots,K\} \right\},$$

où K est le nombre de blocs, $B = (B_1, \dots, B_K)$ est la partition des variables en K blocs, B_k les indices des variables appartenant au bloc k et p_k le nombre de variables dans le bloc k. Parmi les modèles F_B, on distingue l'estimateur du maximum de vraisemblance $\hat{f}_B$:

$$\hat{f}_B = \phi(0, \hat{\mathbf{\Sigma}}_B) \text{ où } \hat{\mathbf{\Sigma}}_B = \begin{pmatrix} \hat{\mathbf{\Sigma}}^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \hat{\mathbf{\Sigma}}^K \end{pmatrix} \text{ et } \hat{\mathbf{\Sigma}}^k = S^k \text{ pour } k \in \{1, \dots, K\}.$$

Une exploration de toutes les partitions de $\mathcal{B}$ n'est pas possible. Comme suggéré par la règle du *screening* diagonale par bloc, nous concentrons notre attention sur le sous-ensemble de partitions suivant, où $\Lambda = \left\{(\hat{\sigma}_{j,j'})_{j>j'}\right\}$:

$$\mathcal{B}^\Lambda = \left\{ B(\lambda), \lambda \in \Lambda \text{ où } B(\lambda) \text{ sont les composantes connexes du graphe } (\mathbf{1}_{\{|\hat{\sigma}_{j,j'}|>\lambda\}})_{jj'} \right\}.$$

On sélectionne la partition B suivant un critère de vraisemblance pénalisée, où $\frac{1}{n}\sum_{i=1}^n \log(\hat{s}_B(\mathbf{y}_i))$ est la log-vraisemblance du modèle et pen un terme de pénalité à définir :

$$\hat{B} = \underset{B\in\mathcal{B}^\Lambda}{\operatorname{argmin}} \left\{ -\frac{1}{n}\sum_{i=1}^n \log(\hat{s}_B(\mathbf{y}_i)) + \text{pen}(B) \right\}. \tag{3.3}$$

Le critère BIC (Schwarz, 1978) correspond au terme de pénalité suivant : $\text{pen}(B) = \frac{\log n}{2} D_B$ où D_B est le nombre de paramètres du modèle F_B. Ce critère est asymptotique consistant : si les données sont issues d'une variable aléatoire de densité $f^\star = \phi(0, \mathbf{\Sigma}_{B^\star})$ alors BIC sélectionne la vraie partition des données $B^\star$ lorsque le nombre d'observations n tend vers l'infini. Dans notre contexte, le nombre d'observations n est limité. Pour cette raison, nous considérons un critère de sélection non-asymptotique basé sur l'heuristique de pente proposé par Birgé and Massart (2007). Cette heuristique consiste à prendre le choix de pénalité suivant :

$$\text{pen}(B) = \kappa D_B,$$

où κ est un coefficient à déterminer. Ce terme de pénalité dépend non seulement du nombre de paramètres définissant chaque modèle mais également de la complexité de la

collection de modèles considérée. Idéalement, on aimerait choisir le modèle qui minimise le risque de Kullback-Leiber avec la vraie distribution de densité $f^\star$. On note $\hat{f}_{B(f^\star)}$ cet estimateur. Il est inconnu car $f^\star$ est inconnue. C'est le modèle oracle que l'on ne peut atteindre en pratique. Mais on peut avoir une procédure qui est telle que l'on contrôle le risque de Kullback-Leibler entre $f^\star$ et le modèle choisi $\hat{f}_{\hat{B}}$. Le modèle sélectionné $\hat{f}_{\hat{B}}$ grâce à cette pénalité vérifie la borne oracle :

$$R(f^\star, \hat{f}_{\hat{B}}) \leq \mathcal{C} R(f^\star, \hat{f}_{B(f^\star)}),$$

où $\mathcal{C}$ est une constante. Nous disposons également d'une borne minimax sur la qualité de l'estimation qui minore le risque $R(f^\star, \hat{f}_{\hat{B}})$ par un terme du même ordre que la borne obtenue dans la borne oracle, à un terme logarithmique près.

Le calcul du coefficient κ peut être effectué explicitement en fonction des conditions imposées sur les paramètres des modèles considérés. En pratique, il est plus simple de calibrer ce paramètre à partir des données à l'aide des deux méthodes détaillées par Baudry et al. (2012). Ces deux méthodes consistent à détecter le coefficient κ à l'aide d'une régression robuste réalisée entre la log-vraisemblance et la dimension du modèle pour les modèles complexes où à l'aide du saut de dimension détecté sur la fonction représentant la dimension du modèle en fonction du paramètre κ à calibrer.

Cette méthode d'inférence de réseaux, ainsi que les résultats théoriques et illustrations sur des données simulées et réelles, sont détaillés au chapitre 8. Ce chapitre fait l'objet d'un article en cours d'écriture avec Emilie Devijver.

Première partie

Pré-traitement des données RNA-seq

Sommaire

Chapitre 4

Data-based filtering for replicated high-throughput transcriptome experiments

Résumé. Le séquençage haut-débit RNA-seq est très utilisé en analyse différentielle des données d'expression des gènes à travers différentes conditions. Les tests sont réalisés sur un grand nombre de gènes ce qui implique un contrôle très stringent du nombre de faux positifs au détriment de la puissance de détection des tests. Afin de modérer l'impact de cette correction pour les tests multiples, les données sont d'abord filtrées afin de retirer de l'analyse les gènes peu exprimés ou présentant peu de signal. Cependant, l'impact de ces filtres sur l'analyse différentielle n'a pas été beaucoup étudié. Nous proposons une méthode de filtrage dont le seuil est calibré à l'aide d'un indice de Jaccard mesurant la similarité entre les différents réplicats d'une même condition, calculé sur les données. Par comparaison avec d'autres méthodes de filtrage régulièrement utilisées sur les données RNA-seq, notre méthode filtre correctement les gènes peu exprimés et contribue à augmenter la puissance de détection des gènes moyennement ou fortement exprimés. De plus, le seuil de filtre, calibré à partir des données, varie en fonction du jeu de données considéré, ce qui souligne l'intérêt de notre méthode. La méthode proposée est implémentée dans le package `R` package `HTSFilter` disponible sur Bioconductor.

Abstract. RNA sequencing is now widely performed to study differential expression among experimental conditions. As tests are performed on a large number of genes, very stringent false discovery rate control is required at the expense of detection power. Ad hoc filtering techniques are regularly used to moderate this correction by removing genes with low signal, with little attention paid to their impact on downstream analyses. We propose a data-driven method based on the Jaccard similarity index to calculate a filtering threshold for replicated RNA-seq data. In comparisons with alternative data filters regularly used in practice, we demonstrate the effectiveness of our proposed method to correctly filter lowly expressed genes, leading to increased detection power for moderately to highly expressed genes. Interestingly, this data-driven threshold varies among experiments, highlighting the interest of the method proposed here. The proposed filtering method is implemented in the `R` package `HTSFilter` available on Bioconductor.

4.1 Introduction

Over the past five years, next-generation high-throughput sequencing (HTS) technology has become an essential tool for genomic and transcriptomic studies. In particular, the use of HTS technology to directly sequence the transcriptome, known as RNA sequencing (RNA-seq), has revolutionized the study of gene expression by opening the door to a wide range of novel applications. Unlike microarray data, which are continuous, RNA-seq data represent highly heterogeneous counts for genomic regions of interest (typically genes), and often exhibit zero-inflation and a large amount of overdispersion among biological replicates ; as such, a great deal of methodological research (e.g., Anders and Huber, 2010; Robinson et al., 2010; Dillies et al., 2013) has recently focused on appropriate normalization and analysis techniques that are adapted to the characteristics of RNA-seq data ; see Oshlack et al. (2010) for a review of RNA-seq technology and analysis procedures.

As with data arising from previous technologies, such as microarrays or serial analysis of gene expression (SAGE), HTS data are often used to conduct differential analyses. In recent years, several approaches for gene-by-gene tests using gene-level HTS data have been proposed, with the most popular making use of Poisson (Wang et al., 2010), overdispersed Poisson (Auer and Doerge, 2011), or negative binomial distributions (Anders and Huber, 2010; Robinson et al., 2010). Because a large number of hypothesis tests are performed for gene-by-gene differential analyses, the obtained p-values must be adjusted to address the fact that many truly null hypotheses will produce small p-values simply by chance ; to address this multiple testing problem, several well-established procedures have been proposed to adjust p-values in order to control various measures of experiment-wide false positives, such as the false discovery rate (FDR). Although such procedures may be used to control the number of false positives that are detected, they are often at the expense of the power of an experiment to detect truly differentially expressed (DE) genes, particularly as the number of genes in a typical HTS dataset may be in the thousands or tens of thousands. To reduce this impact, several authors in the microarray literature have suggested the use of data filters in order to identify and remove genes which appear to generate an uninformative signal (Bourgon et al., 2010) and have no or little chance of showing significant evidence of differential expression ; only hypotheses corresponding to genes that pass the filter are subsequently tested, which in turn tempers the correction needed to adjust for multiple testing.

In recent work, Bourgon et al. (2010) advocate for the use of *independent data filtering*, in which the filter and subsequent test statistic pairs are marginally independent under the null hypothesis and the dependence structure among tests remains largely unchanged pre- and post-filter, ensuring that post-filter p-values are indeed true p-values. For such an independent filter to be effective, it must be positively correlated with the test statistic under the alternative hypothesis ; indeed, it is this correlation that leads to an increase in detection power after filtering. In addition, Bourgon *et al.* demonstrate that non-independent filters for which dependence exists between the filter and test statistic (e.g., making use of condition labels to filter genes with average expression in at least one condition less than a given threshold), can in some cases lead to a loss of control of experiment-wide error rates.

Several ad hoc data filters for RNA-seq data have been used in recent years, including

filtering genes with a total read count smaller than a given threshold (Sultan et al., 2008) and filtering genes with at least one zero count in each experimental condition (Bottomly et al., 2011) ; however, selecting an arbitrary threshold value to filter genes in this way does not account for the overall sequencing depth or variability of a given experiment. One exception to these ad hoc filters is the work of Ramsköld et al. (2009), in which a comparison between expression levels of exonic and intergenic regions was used to find a threshold for detectable expression above background in various human and mouse tissues, where expression was estimated as Reads Per Kilobase per Million mapped reads (RPKM) (Mortazavi et al., 2008). The threshold of 0.3 RPKM identified in this work has in turn been applied to several other studies (e.g., Łabaj et al., 2011; Cánovas et al., 2010; Sam et al., 2011). However, to our knowledge, although filters for read counts are routinely used in practice, little attention has been paid to the choice of the type of filter or threshold used or its impact on the downstream analysis.

In this paper, we propose a novel data-based procedure to choose an appropriate filtering threshold based on the calculation of a similarity index among biological replicates for read counts arising from replicated high-throughput transcriptome sequencing data. This technique provides an intuitive data-driven way to filter RNA-seq data and to effectively remove genes with low, constant expression levels. Our proposed filtering threshold may be useful in a variety of applications for RNA-seq data, including differential expression analyses, clustering and co-expression analyses, and network inference.

4.2 Methods

4.2.1 Types of filters used for RNA-seq data

Data filters are routinely used in practice for differential analyses of RNA-seq data. Most such filters are applied to data that have been normalized in some way, rather than directly to the raw counts, in order to account for systematic inter-sample biases typical of RNA-seq data, e.g., differences in library size (Robinson and Oshlack, 2010; Anders and Huber, 2010) or GC content (Risso et al., 2011; Hansen et al., 2012). In particular, the Trimmed M-Means (TMM) library size normalization (Robinson and Oshlack, 2010) and the normalization included in the `DESeq` Bioconductor package (Anders and Huber, 2010) have been found to be robust methods to correct for library size biases, even in the presence of widely different library compositions (Dillies et al., 2013).

We consider two broad categories of filters for RNA-seq data, based on the filtering criterion used : mean-based filters and maximum-based filters. We note that although variance-based filters are routinely used for microarray data (Bourgon et al., 2010), they have not been applied to RNA-seq data ; this is likely due to the small number of replicates available in most RNA-seq datasets (and thus, the difficulty in obtaining accurate estimates of per-gene variances) and the fact that the variance is assumed to be a function of the mean under a negative binomial model.

Mean-based filters

In mean-based filters, genes with mean normalized counts across all samples less than or equal to a pre-specified cutoff are filtered from the analysis. Some authors (Sultan et al., 2008) have also proposed filtering genes with a total read count less than or equal to a given threshold s ; we note that this is equivalent to mean-based filters for threshold s divided by the number of samples.

In addition to normalized counts, we also consider mean-based filters for the RPKM (Mortazavi et al., 2008) measure, which was initially proposed to simultaneously normalize RNA-seq data for biases due to library size and gene length. However, we note that it has been shown that counts, rather than RPKM values, are preferable for the differential analysis of RNA-seq data (Oshlack and Wakefield, 2009). For this reason, after filtering genes with a RPKM mean filter, raw counts are used for the subsequent differential analysis. A comparison of differential analysis methods developed for counts and RPKM values is beyond the scope of this work.

Maximum-based filters

In maximum-based filters, genes with maximum normalized counts across all samples less than or equal to a pre-specified threshold are filtered from the analysis. As above, in addition to normalized counts we also consider maximum-based filters for RPKM values, which we refer to as a RPKM maximum filter.

A generalization of the maximum-based filter has also been proposed in the `edgeR` analysis pipeline (Robinson et al., 2010) based on counts per million (CPM), calculated as the raw counts divided by the library sizes and multiplied by one million. Genes with a CPM value less than a given cutoff (e.g., 1 or 100) in more samples (ignoring condition labels) than the size of the smallest group are subsequently filtered from the analysis. To distinguish this approach from the other maximum-based filters, we refer to this strategy as a CPM filter.

We note that maximum-based filters are not independent filters as described by Bourgon et al. (2010) ; in particular, for extremely large filtering thresholds, maximum-based filters do not guarantee control of the Type I error rate if p-values are computed using the pre-filter null distribution. For the threshold values typically used in practice (e.g., based on a quantile, or the data-based threshold proposed below), this is usually not a concern (see Supplementary [1] Figure 28). Although it may be difficult to verify that conditional and unconditional p-value distributions coincide for real data, it may be useful to examine histograms of each (for example, as shown in Figure 4.3).

4.2.2 A data-based threshold for maximum-based filters

For each of the filter types previously defined, a biologically pertinent cutoff (or alternatively, number of genes to be filtered) must be chosen ; in practice, arbitrary thresholds are routinely used with little or no discussion of their impact on the downstream analysis. To address this issue, we propose a data-based choice for the threshold

1. http ://bioinformatics.oxfordjournals.org/content/suppl/2013/06/24/btt350.DC1/Full$_{S}uppMat.pdf$

		Sample j	
		Normalized counts $> s$	Normalized counts $\leq s$
Sample j'	Normalized counts $> s$	a	b
	Normalized counts $\leq s$	c	d

TABLE 4.1 – Definition of the constants used to calculate the Jaccard similarity index for a pair of samples j and j' and a given threshold s. The constant a represents the number of genes with normalized counts greater than s in both samples j and j', and so on.

to be used in maximum-based filters. The main idea underlying this choice is to identify the threshold that maximizes the filtering similarity among replicates, that is, one where most genes tend to either have normalized counts less than or equal to the cutoff in all samples (i.e., filtered genes) or greater than the cutoff in all samples (i.e., non-filtered genes).

To define this filtering similarity, we begin with some notation. Let y_{gj} represent the observed normalized read count (e.g., after scaling raw counts by library size) for gene g in sample j and let $\mathcal{C}(j)$ represent the experimental condition of sample j, with $g \in \{1, \ldots, G\}$ and $j \in \{1, \ldots, J\}$. Typically, in the context of differential analyses, the number of conditions is equal to 2. We denote the full vector of read counts in a given sample as $\mathbf{y}_j$. We now wish to define a *similarity index* between a pair of replicates within the same condition $\{(\mathbf{y}_j, \mathbf{y}_{j'}) : \mathcal{C}(j) = \mathcal{C}(j')\}$ after binarizing the data for a fixed cutoff s (1 if $y_{gj} > s$ and 0 otherwise). We note that a variety of similarity indices have been proposed since the early 1900s ; however, in a comparison among a set of similarity indices (see the Supplementary Materials [2]) we found the Jaccard index (Jaccard, 1901) to be simple, natural, and easy to interpret for the analysis of high-throughput sequencing data. This index is defined as follows :

$$J_s(\mathbf{y}_j, \mathbf{y}_{j'}) = \frac{a}{a + b + c} \tag{4.1}$$

where a, b, and c are defined in Table 4.1. We note that $J_s(\mathbf{y}_j, \mathbf{y}_{j'})$ takes on values from 0 (dissimilar) to 1 (similar). Because multiple replicates and/or conditions are typically available in HTS experiments, we extend the definition of the pairwise Jaccard index in Equation (4.1) to a global Jaccard index by averaging the indices calculated over all pairs in each condition :

$$J_s^\star(\mathbf{y}) = \text{mean}\left\{J_s(\mathbf{y}_j, \mathbf{y}_{j'}) : j < j' \text{ and } \mathcal{C}(j) = \mathcal{C}(j')\right\}. \tag{4.2}$$

Using the global Jaccard index defined in Equation (4.2) as a measure of similarity, we now wish to identify the cutoff $s^\star$ for normalized counts that corresponds to the

2. http ://bioinformatics.oxfordjournals.org/content/suppl/2013/06/24/btt350.DC1/Full$_S uppMat.pdf$

greatest similarity possible among replicates, that is, the value of s corresponding to the maximum value of the global Jaccard index :

$$s^\star = \underset{s}{\operatorname{argmax}}\, J_s^\star(\mathbf{y}). \tag{4.3}$$

In practice, for the calculation of the data-based global filtering threshold in Equation (4.3), we calculate the value of the global Jaccard index in Equation (4.2) for a fixed set of threshold values and fit a loess curve (Cleveland, 1979) through the set of points ; the value of $s^\star$ is subsequently set to be the maximum of these fitted values.

Once the data-driven filter threshold for normalized counts $s^\star$ has been identified, the subsequent steps to be taken may change for different applications. To perform an analysis of differential expression between two experimental conditions, we propose using this threshold $s^\star$ in a maximum-based filter, as defined in Section 4.2.1 ; in the following, we refer to this technique as the *Jaccard filter*.

4.2.3 The Bioconductor package `HTSFilter`

The proposed filtering method is implemented in the `HTSFilter` package, currently available as part of the Bioconductor project (Gentleman et al., 2004) within the statistical environment `R` (R Development Core Team, 2009). The `HTSFilter` package is compatible with a variety of data classes and analysis pipelines, including `matrix` and `data.frame` objects, the S4 class `CountDataSet` in the `DESeq` pipeline (Anders and Huber, 2010), and the S3 class `DGEList` in the `edgeR` pipeline (Robinson et al., 2010). A package vignette describes the use of the `HTSFilter` package within each of these pipelines.

4.3 Results

In the following, we apply the normalization approach proposed by Anders and Huber (2010) for mean- and maximum-based filters, although other types of normalization may be appropriate for some data. For gene-by-gene comparisons between two conditions, we illustrate the use of the proposed filter in conjunction with the model proposed in the `DESeq` Bioconductor package (version 1.8.3), which has been developed in order to model count data with a small number of replicates in the presence of overdispersion (Anders and Huber, 2010) ; p-values are adjusted for multiple testing using the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) to control the false discovery rate (FDR). We note that the filtering method proposed here may also be used in conjunction with other popular methods, e.g., `edgeR` (Robinson et al., 2010). See the Supplementary Materials for additional discussion of the normalization and statistical testing methods used in this work.

In practice, there may be some question about the appropriate point in the analysis pipeline to apply data filters : Should normalized data first be filtered, then normalization factors re-estimated and the model fit (i.e., mean and dispersion parameters estimated) ? Should normalization factors and model parameters be estimated based on the full data, and the data filtered only at the end of the analysis pipeline ? The difference

between the two options is nontrivial, particularly as the differential analysis approaches implemented in the `DESeq` and `edgeR` packages both borrow information across genes (whether all or only those passing the filter) to obtain per-gene parameter estimates. In this work, we present results based on the application of filters applied as late in the pipeline as possible, i.e., after library size and dispersion parameter estimation ; a more detailed discussion of this issue is included in the Supplementary Materials.

4.3.1 Description of data

We applied our proposed Jaccard index filter, in addition to the alternative filter types described above, on the following data :

— **Sex-specific expression of liver cells in human.** Sultan et al. (2008) obtained high-throughput transcriptome sequencing data from a human embryonic kidney and a B cell line, with two biological replicates each. The raw read counts and phenotype tables were obtained from the ReCount online resource (Frazee et al., 2011).

— **Differential striatal expression between inbred mouse strains**. Bottomly et al. (2011) performed RNA-seq experiments for ten biological replicates of the C57BL/6J inbred mouse strain and eleven for the DBA/2J strain, and the results were compared with those arising from two different microarray platforms. The raw read counts and phenotype tables were obtained from the ReCount online resource (Frazee et al., 2011).

— **MiTF repression in a human melanoma cell line**. Strub et al. (2011) obtained high-throughput sequencing data to compare gene expression in a melanoma cell line expressing the Microphtalmia Transcription Factor (MiTF) to one in which small interfering RNAs (siRNAs) were used to repress MiTF, with three biological replicates in each group. The raw read counts and phenotype tables are available in the Supplementary Materials of Dillies et al. (2013).

— **Simulated data**. To investigate the effect of the various filtering methods on downstream results, we developed a simulation framework as described in Section 4.3.3.

For the three real datasets (described in further detail in Supplementary Table 1), gene annotations for *Mus musculus* (NCBIM37) and *Homo sapiens* (GRCh37.p7) were obtained from Ensembl version 67 (Birney et al., 2004) using the Biomart tool (Kasprzyk et al., 2004), and the length of each gene in base pairs was calculated. In the Bottomly, Sultan, and Strub data, about 4%, 2%, and 5% (respectively) of the genes had been retired from Ensembl ; for these genes, RPKM-based filters were not used. Readers wishing to examine the complete analyses in detail may find a Sweave document containing commented `R` code for the analyses of each of the datasets in the Supplementary Materials.

4.3.2 Comparison of filters on real data

For both real datasets, differential analyses were performed using a negative binomial model (Anders and Huber, 2010) for unfiltered data and after filtering the data using the techniques described above. For the Jaccard index filter, the global Jaccard index

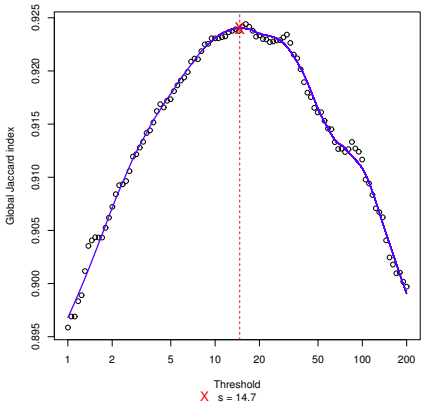

FIGURE 4.1 – Global Jaccard index for the Bottomly data calculated for a variety of threshold values for normalized counts, with a loess curve (blue line) superposed and data-driven threshold value (red cross and red dotted line) equal to $s^\star = 14.7$.

in Equation (4.2) was calculated for a range of threshold values s, yielding a largely unimodal distribution of values for all datasets considered (Figure 4.1 and Supplementary Figures 11 and 18). We also note that the data-driven threshold values $s^\star$ identified for the Bottomly, Sultan, and Strub data were not equal ; in the case of the Bottomly data, the threshold for normalized counts was found to be 14.7, while this threshold was found to be 11.5 for the Sultan data and 103.5 for the Strub data. These differences in filtering threshold among experiments are due to both sequencing depth and variability within the data ; in particular, experiments with greater sequencing depth will tend to have higher filtering thresholds, and those with greater variability will tend to have lower filtering thresholds. Among the data considered in our study, the Strub data have the highest sequencing depth (1.5×10^8) coupled with low intra-condition variability (minimum correlation among replicates equal to 0.98), and they also have the highest threshold considered here. On the other hand, the Sultan data have a much lower sequencing depth (1.8×10^6), and thus have a much lower threshold.

For each dataset, the Jaccard filter was applied with the corresponding data-based threshold calculated above. For the alternative mean- and maximum-based filters for normalized counts and RPKM values, cutoffs were chosen based on the 15% quantile of the respective criterion. For the CPM filter, as suggested in the `edgeR` pipeline Robinson et al. (2010), genes with a CPM value less than 1 in more samples than the size of the smallest group are subsequently filtered from the analysis.

Among the filters considered, it may immediately be seen in Figure 4.2 that in the Strub data, with the exception of the CPM and Jaccard filters, most of the filters considered here appear to be ineffective as they are largely unable to filter genes with very low levels of expression and small log-fold changes ; a similar phenomenon may be observed in the Bottomly and Sultan data (Supplementary Figures 7 and 14). Although these techniques are thus unlikely to (incorrectly) filter truly DE genes, the number of statistical tests is not markedly reduced, and as such the power to detect differential expression will remain largely unchanged as compared to the unfiltered data. With the exception of the RPKM-based filters, all are able, to some extent, to identify and remove genes contributing to a peak of raw p-values close to one (Figure 4.3 and Supplementary Figures 13 and 20), a phenomenon due to the discretization of p-values for small counts ; indeed, histograms of raw p-values following the application of most filters appear to be

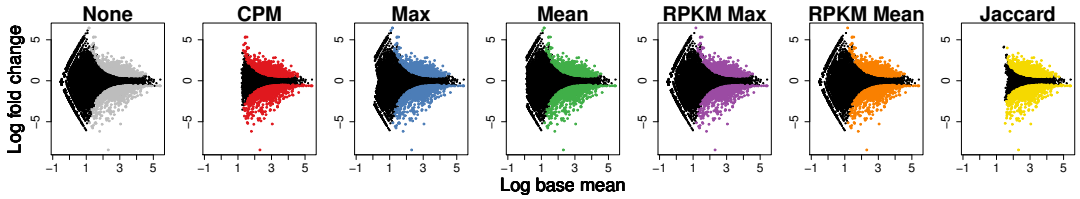

FIGURE 4.2 – Log mean expression versus log fold change values for the Strub data. For each filter, genes identified as non-differentially and differentially expressed are drawn in black and colors, respectively, and those filtered from the differential analysis are omitted from the plot. From left to right, the filters are as follows : none, CPM, maximum, mean, RPKM maximum, RPKM mean, and maximum using the global Jaccard index threshold.

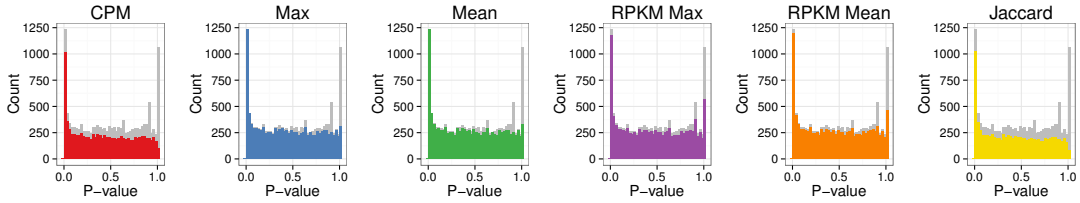

FIGURE 4.3 – Histograms of raw p-values from a differential analysis of the Bottomly data for a variety of filter types. Histograms in grey in the background represent the raw p-values from a differential analysis of the Bottomly data using unfiltered data ; histograms in color in the foreground represent the raw p-values from a differential analysis of the data filtered with various filter types. Figure made using the `ggplot2` package (Wickham, 2009).

roughly uniformly distributed under the null hypothesis. In addition, we note that the proposed Jaccard filter is able to more effectively remove genes with moderate log base means and small log fold changes (see Figure 4.2). Similar conclusions may be drawn from the volcano plots shown in Supplementary Figures 8, 15, and 22.

It is also of interest to consider the effect of each filter on the number of DE genes identified at various levels of expression ; in Figure 4.4 and Supplementary Figures 5 and 12, we note that in all data sets the Jaccard filter leads to more discoveries at all but very weak levels of expression (i.e., mean expression less than 10), with this difference being particularly marked for moderate levels of expression (i.e., mean expression greater than 50). We note that a large number of the missed discoveries for the Jaccard filter at very low levels of expression correspond to genes with zero read counts in one condition and a small number of read counts in the other ; for example, in the Sultan data 50.3% of the 449 discoveries among genes with mean normalized read counts less than 10 had zero read counts in one of the two conditions. Among the other filter types, the CPM filter appears to come closest to the Jaccard filter, with the remaining filters performing similarly.

In considering the overlap of genes filtered using each method (Supplementary Figures 9, 16, and 23), it is clear that a large number of genes may be filtered regardless of the technique used. However, the Jaccard index is better able to filter a large number of weakly expressed genes in all three of the datasets considered here, leading to a more moderate correction for multiple testing ; the direct consequence of this is a larger

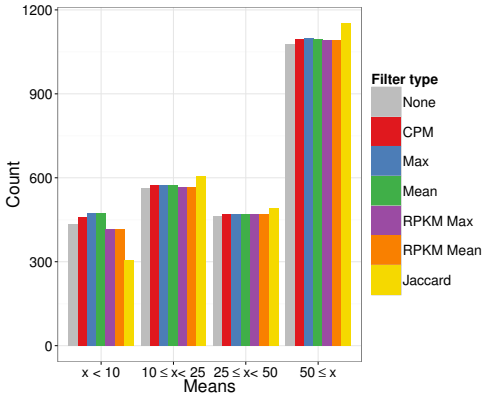

FIGURE 4.4 – Number of DE genes detected in the Sultan data, categorized by normalized base mean for each filter type. Figure made using the `ggplot2` package (Wickham, 2009).

number of discoveries at moderate to high levels of expression. In order to determine whether this advantage is due to the filtering type (i.e., maximum) or the threshold used for each (i.e., using the 15% quantile or the data-based threshold identified by the global Jaccard index), we consider a set of simulation studies in the next section.

4.3.3 Simulated data

Data were simulated using a negative binomial model, with parameters chosen based on the Bottomly, Sultan, and Strub datasets. Briefly, for each dataset, genes with zero counts in one of the two groups and mean less than 5 were removed (see Supplementary Table 1). Differential analyses were performed on unfiltered data using the `DESeq` Bioconductor package (Anders and Huber, 2010) as described in the Supplementary Materials. After adjusting raw p-values for multiple testing using the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995), we identified 570, 2515, and 2485 differentially expressed genes, respectively, at the 5% significance level in the Bottomly, Sultan, and Strub data. In addition, the parametric gamma regressions fitted to the per-gene dispersion estimates α and gene expression means μ (Supplementary Figure 25) were identified for each dataset :

$$\alpha_{\text{Bottomly}}(\mu) = 0.03 + 0.72/\mu, \tag{4.4}$$

$$\alpha_{\text{Sultan}}(\mu) = 0.01 + 1.23/\mu, \tag{4.5}$$

$$\alpha_{\text{Strub}}(\mu) = 0.03 + 13.35/\mu. \tag{4.6}$$

Subsequently, simulation parameters for each dataset were fixed as follows. For genes identified as being differentially expressed, means for each condition were set to be the empirically calculated means from each condition from the normalized data ; for genes not identified as being differentially expressed, means for each condition were both set to be the global mean (across both conditions) from the normalized data. Note that this allows genes to be simulated as differentially expressed across the full range of mean expression values (Supplementary Figure 27). Per-gene dispersion parameters were set to be the fitted values from the regression equations defined in Equations (4.4)-(4.6) as a function of the overall mean for each gene ; for the simulations based on the

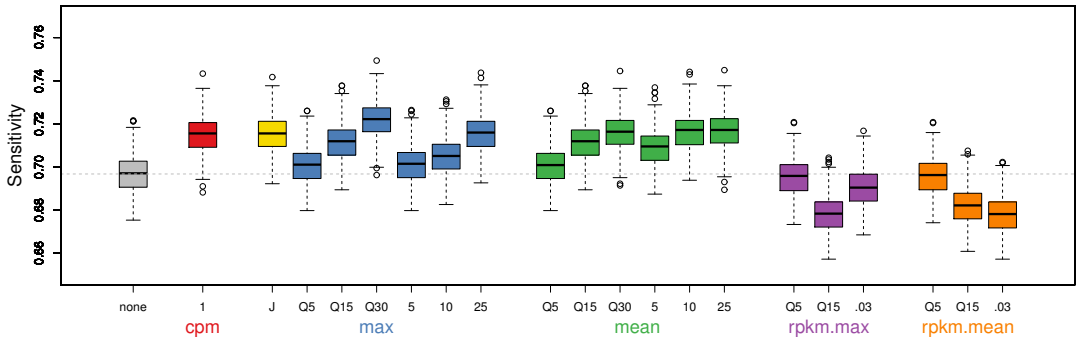

FIGURE 4.5 – Sensitivity (over simulated 300 datasets) to detect differentially expressed genes for a variety of filter types and cutoffs, with simulation parameters based on the Strub data. None : no filter. CPM : genes with a CPM less than one in more than half the samples are filtered. Max : maximum-based filter, using the threshold based on the Jaccard index (J), quantiles (5%, 15%, 30%), or values (5, 10, 25). Mean : mean-based filter, using the threshold based on quantiles (5%, 15%, 30%), or values (5, 10, 25). RPKM.max : maximum RPKM filter, using the threshold based on quantiles (5%, 15%) or the value 0.3. RPKM.mean : maximum RPKM filter, using the threshold based on quantiles (5%, 15%) or the value 0.3.

Bottomly and Sultan data, dispersion parameters for genes with overall mean expression less than 20 were fixed to be equal to 10^{-10} to simulate negligible overdispersion, as shot noise appears to dominate biological noise at low expression levels in these data (Supplementary Figure 26). Once these parameters were fixed for each gene, a negative binomial model was used to simulate 300 individual datasets each for the parameters based on the Bottomly, Sultan and Strub data, with 21 samples (10 in one condition and 11 in the other), 4 samples (2 in each condition), and 6 samples (3 in each condition), respectively. Real lengths corresponding to the genes in each dataset were used for the calculation of RPKM values.

4.3.4 Comparison of filters on simulated data

In order to assess performance on simulated data, we focus on the sensitivity of detecting differentially expressed genes after each data filter, defined as the proportion of truly DE genes detected among all truly DE genes. In addition, we construct Receiver Operating Characteristic (ROC) curves of each filter, based on the *filtering sensitivity*, defined as the proportion of correctly unfiltered genes (i.e., DE and unfiltered) among all truly DE genes, and the *filtering specificity*, defined as the proportion of correctly filtered genes (i.e., non-DE and filtered) among all non-DE genes.

In Figure 4.5, we note that the sensitivity to detect differentially expressed genes greatly varies among the filtering types for simulations based on the Strub data, as well as among different thresholds within each filtering type ; in addition, the RPKM maximum and RPKM mean filters actually lead to lower detection power than unfiltered data. Similar results may be seen for the simulations based on the Bottomly and Sultan data in Supplementary Figure 30. For the simulation setting shown in Figure 4.5, larger thresholds appear to yield the highest detection sensitivity (i.e., maximum or mean-based filters for normalized counts using the 30% quantile as a threshold). However, this

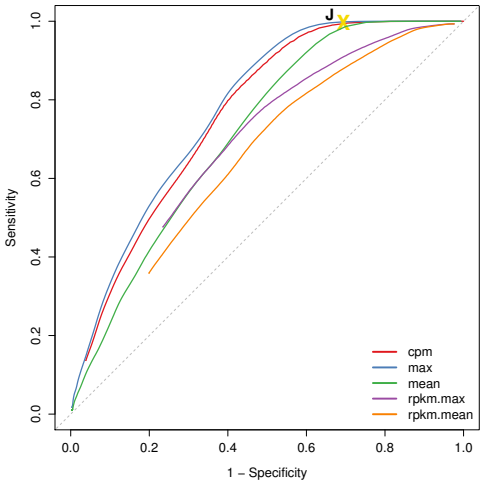

FIGURE 4.6 – ROC curves (averaged over 300 datasets) for the filtering performance on simulated data based on the Sultan data for the CPM, maximum, mean, maximum RPKM and mean RPKM filters over a range of cutoffs. The yellow cross labeled with a "J" corresponds to the filtering sensitivity and specificity for the data-based threshold chosen via the global Jaccard index.

trend is reversed for the other simulation settings, where smaller cutoffs (i.e., cutoffs of 5% or 15% for the maximum or mean-based filters) lead to higher detection sensitivity. This highlights the difficulty in pre-selecting a fixed threshold for a given filtering method, as well as the advantage of our proposed Jaccard filter. Indeed, the Jaccard filter appears to lead to high detection sensitivity for all simulations with the exception of those based on the Bottomly data ; for these data, because many weakly expressed genes were simulated to be differentially expressed (see Supplementary Figure 27), unfiltered data had the highest sensitivity.

To assess the role of the choice of threshold for each filter type, we constructed ROC curves of the filtering sensitivity and specificity over varying cutoffs in Figure 4.6 and Supplementary Figure 29. We note that the mean and maximum RPKM-based filters tend to have lower filtering sensitivity than the others, and the maximum filters (for both normalized counts and CPM values) tend to be quite similar ; however, for all simulation settings, the maximum filter for normalized counts has a slight advantage over that for CPM values. In other words, regardless of the threshold used for filtering, maximum-based filters appear to more effectively filter non-DE genes than the remaining methods. Finally, we note also that the data-based threshold using the global Jaccard index appears to find a good compromise between filtering sensitivity and filtering specificity.

4.4 Conclusions and discussion

Data filtering has proven to be of great practical importance for the differential analysis of high-throughput microarray and RNA-seq data by identifying and removing genes with uninformative signal prior to testing. In recent years, many ad hoc procedures have been used to filter RNA-seq data, such as filtering genes with a total or mean normalized read count less than a specified threshold. However, despite its impact on

the downstream analyses, no clear recommendations have yet been provided concerning the choice of filtering technique.

Among the filter types considered here, we have found that filters using the maximum normalized count appear to be best able to correctly filter genes with low levels of expression and little evidence of differential expression. In addition, we have proposed a method to calculate a data-driven and non pre-fixed filtering threshold value for normalized counts from replicated RNA-seq data, based on the global Jaccard similarity index. In particular, our proposed filtering technique was found to remove from the analysis a large number of genes with little or no chance of showing evidence of differential expression, and therefore to increase detection power at moderate to high levels of expression through a moderation of the correction for multiple testing. As such, we recommend that genes with a normalized count value less than this data-driven threshold in all samples be filtered from subsequent differential analyses. We emphasize that the data-driven threshold value may vary greatly among RNA-seq experiments due to differences in sequencing depth and intra-condition variability (see Supplementary Figure 31 for the data-based thresholds calculated on three additional RNA-seq datasets) ; as such, the threshold value must be recalculated for each data set of interest.

The impact of the proposed filtering method has been investigated here in the context of differential analyses. We anticipate that it will also be useful in a variety of other applications, for example detecting genes that are specifically expressed in one condition or ubiquitously expressed across several conditions, which is often a crucial biological question. In addition, we anticipate that such filtering will be useful, for example, in co-expression or network reconstruction analyses to remove genes with low, constant levels of expression. Finally, we note that although this filter was presented here for the analysis of RNA-seq data, it can readily be applied to other types of replicated high-throughput sequencing data, such as CHIP-seq data.

Deuxième partie

Classification par modèle de mélange

Sommaire

Chapitre 5

Transformation des données et comparaison de modèles pour la classification des données RNA-seq

Résumé. Les données d'expression issues du séquençage haut-débit (RNA-seq) sont des données de comptage très hétérogènes. Il est naturel de les représenter par des modèles basés sur des lois discrètes comme la loi de Poisson ou la loi binomiale négative. Mais des transformations simples des données peuvent permettre de se ramener à des modèles plus répandus fondés sur des lois gaussiennes. Nous montrons comment comparer objectivement les vraisemblances de ces modèles travaillant sur des données différentes. Nous nous focalisons pour mener ces comparaisons sur des problèmes de classification où les mélanges de Poisson et gaussiens peuvent être mis en compétition.

Abstract. High-throughput transcriptome sequencing data (RNA-seq) are made up of highly heterogeneous counts. Although they are often modeled with discrete distributions, including the Poisson and negative binomial distributions, Gaussian models on transformed data could alternatively be considered. We show how the likelihood of these different models can be objectively compared. We focus attention on the problem of clustering gene profiles, where Poisson mixtures on count data are compared with Gaussian mixtures on transformed data.

5.1 Les modèles de mélange pour la classification des données RNA-seq

5.1.1 Modèle de mélange de lois de Poisson

Rau et al. (2015) ont proposé une paramétrisation du modèle de mélange de lois de Poisson spécifique aux données RNA-seq. Cette paramétrisation prend en compte les différentes conditions expérimentales avec des réplicats biologiques. Nous considérons une version simplifiée de cette modélisation sans prendre en compte les réplicats par condition afin d'alléger les notations.

On dispose d'une matrice de mesures d'expression de n gènes $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$. Pour chaque gène i ($i = 1, \ldots, n$), le vecteur $\mathbf{y}_i$ indique l'expression du gène i pour les p conditions expérimentales j ($j = 1, \ldots, p$). On suppose que les données $\mathbf{y}$ sont la réalisation d'un mélange de K variables aléatoires de lois de Poisson de densité :

$$f(\mathbf{y}_i; K, \theta_K) = \sum_{k=1}^{K} p_k \prod_{j=1}^{p} \mathcal{P}(y_{ij}; \mu_{ijk}). \tag{5.1}$$

Suivant Rau et al. (2015), les paramètres $(p_1, \ldots, p_K)$ sont les proportions de chaque composante du mélange et $\prod_{j=1}^{p} \mathcal{P}(y_{ij}; \mu_{ijk})$ est la densité d'un vecteur de p variables aléatoires indépendantes de lois de Poisson de moyennes respectives $\mu_{ijk} = w_i s_j \lambda_{jk}$ pour $k = 1, \ldots, K$. On note $y_{i\cdot} = \sum_{j=1}^{p} y_{ij}$, $y_{\cdot j} = \sum_{i=1}^{n} y_{ij}$ et $y_{\cdot\cdot} = \sum_{i=1}^{n} \sum_{j'=1}^{p} y_{ij'}$. Les facteurs $w_i = y_{i\cdot}$ prennent en compte le niveau d'expression de chaque gène, ils sont calculés avant tout estimation des paramètres du modèle. Les facteurs s_j corrigent le biais technique spécifique aux données RNA-seq lié aux différences de profondeur de séquençage entre échantillons (voir Chapitre 1). L'expression $s_j = \frac{y_{\cdot j}}{y_{\cdot\cdot}}$ correspond à un estimateur du maximum de vraisemblance. Les paramètres $\boldsymbol{\lambda}_k = (\lambda_{1k}, \ldots, \lambda_{pk})$ correspondent aux profils d'expression des gènes de la composante k. Ainsi, le modèle classe les gènes en fonction de leur dynamique d'expression $(\boldsymbol{\lambda}_k)$ et non en fonction de leur niveau d'expression absolu (w_i). Les paramètres p_k et λ_{jk} sont estimés par l'algorithme EM sous les contraintes $\sum_{k=1}^{K} p_k = 1$ et $\sum_{j=1}^{p} \lambda_{jk} s_j = 1$ pour tout k. L'implémentation de l'estimation des paramètres de ce modèle est proposée dans le package `HTSCluster`.

5.1.2 Transformation des données RNA-seq et modèle de mélange de lois gaussiennes

Une alternative à ce modèle de mélange de lois de Poisson est un modèle de mélange de lois gaussiennes, classiquement utilisé pour les données de puces à ADN. Dans un cadre différent, celui de l'analyse différentielle d'expression de gènes, Law et al. (2014) ont proposé une transformation logarithmique des données RNA-seq afin d'utiliser les modèles linéaires gaussiens développés initialement pour l'analyse des données de puces. Dans le même esprit, on propose ici le même type de transformation pour l'utilisation d'un modèle de mélange de lois gaussiennes.

Les données $\mathbf{y}$ sont transformées de sorte que l'objectif de classification reste le plus proche de celui du modèle de mélange de Poisson précédent (modélisation de la dy-

namique d'expression entre conditions). Chaque comptage y_{ij} est divisé par le facteur $N_j = \frac{y_{\cdot j}}{10^6}$ afin de corriger le biais technique spécifique aux données RNA-seq lié aux différences de profondeur de séquençage entre échantillons (voir Chapitre 1). Le facteur N_j est le nombre comptages pour condition j, exprimé en million. Il correspond globalement au facteur s_j du modèle de mélange de lois de Poisson. Afin de modéliser la variation d'expression du gène, on compare le comptage normalisé y_{ij}/N_j à $m_i = \frac{1}{p}\sum_{j'=1}^{p} \frac{y_{ij'}}{N_{j'}}$, l'expression moyenne du gène i à travers les p conditions expérimentales. Le facteur m_i correspond globalement au facteur w_i dans le modèle de mélange de lois de Poisson. Une constante est ajoutée au numérateur et au dénominateur pour éviter les problèmes numériques. On nomme cette transformation des données t :

$$t(y_{ij}) = \log\left(\frac{y_{ij}/N_j + 1}{m_i + 1}\right).$$

On modélise le vecteur des données transformées $\tilde{\mathbf{y}}_i = t(\mathbf{y}_i)$ par un mélange de K lois gaussiennes de densité :

$$g(\tilde{\mathbf{y}}_i; K, \eta_k) = \sum_{k=1}^{K} p_k \phi(\tilde{\mathbf{y}}_i; \boldsymbol{\nu}_k, \Sigma_k). \tag{5.2}$$

Les paramètres $(p_1, \ldots, p_K)$ sont les proportions de chaque composante du mélange et $\phi(\tilde{\mathbf{y}}_i; \boldsymbol{\nu}_k, \Sigma_k)$ est la densité d'une loi normale de dimension p de moyenne $\boldsymbol{\nu}_k$ et de variance-covariance Σ_k. Une implémentation de l'estimation des paramètres de ce modèle est proposée dans le package `Rmixmod` (Lebret et al., 2013).

5.2 Transformation des données et comparaison de modèles

La vraisemblance du modèle de mélange de lois de Poisson sur les données brutes par rapport à la mesure de comptage s'écrit :

$$\ell_f(\mathbf{y}_1, \ldots, \mathbf{y}_n; K, \theta_K) = \prod_{i=1}^{n} f(\mathbf{y}_i; K, \theta_K).$$

La vraisemblance du modèle de mélange gaussien sur les données transformées par rapport à la mesure de Lebesgue s'écrit :

$$\ell_g(\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_n; K, \eta_K) = \prod_{i=1}^{n} g(\tilde{\mathbf{y}}_i; K, \eta_k).$$

La comparaison de ces deux modèles est *a priori* difficile car les vraisemblances des modèles s'écrivent par rapport à des mesures différentes. Cependant, la densité du modèle de Poisson peut être définie comme la limite d'une densité par rapport à la mesure de Lebesgue. Pour tout $x \in \mathbb{R}$ et $0 < \Delta < \frac{1}{2}$, on considère :

$$f_\Delta(x) = \begin{cases} \left(\exp(-\lambda)\frac{\lambda^t}{t!}\right)\frac{1}{\Delta}, & \text{si } x \in]t - \frac{\Delta}{2}; t + \frac{\Delta}{2}[, t \in \mathbb{N} \\ 0, & \text{sinon.} \end{cases}$$

Cette densité est bien définie par rapport à la mesure de Lebesgue et sa limite, lorsque Δ tend vers 0, est la densité d'une loi de Poisson. Dans ce contexte, on peut écrire l'égalité des densités $d\tilde{\mathbf{y}}_i = t'(\mathbf{y}_i)d\mathbf{y}_i$ à partir de la relation $\tilde{\mathbf{y}}_i = t(\mathbf{y}_i)$. On obtient la relation suivante :

$$g(\tilde{\mathbf{y}}_i; K, \eta_k)d\tilde{\mathbf{y}}_i = g(t(\mathbf{y}_i); K, \eta_k)t'(\mathbf{y}_i)d\mathbf{y}_i.$$

Cette relation permet de réécrire la vraisemblance du modèle de mélange sur données transformées en fonction des données initiales $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$:

$$\ell_g(\mathbf{y}_1, \ldots, \mathbf{y}_n; K, \eta_K) = \prod_{i=1}^{n} g(t(\mathbf{y}_i); K, \eta_k)t'(\mathbf{y}_i).$$

Les deux modèles peuvent alors être comparés par un critère de vraisemblance pénalisée comme le BIC :

$$\mathrm{BIC}_f(K; \mathbf{y}) = \sum_{i=1}^{n} \log f(\mathbf{y}_i; K, \hat{\theta}_K) - \frac{\nu_f}{2}\log(n),$$

$$\mathrm{BIC}_g(K; \mathbf{y}) = \sum_{i=1}^{n} \log g(t(\mathbf{y}_i); K, \hat{\eta}_k) + \sum_{i=1}^{n} \log t'(\mathbf{y}_i) - \frac{\nu_g}{2}\log(n).$$

Les quantités $\hat{\theta}_K$ et $\hat{\eta}_K$ sont les estimateurs du maximum de vraisemblance des paramètres des modèles respectifs, ν_f et ν_g sont les nombres de paramètres des modèles respectifs. Le modèle s'ajustant le mieux aux données est le modèle maximisant le critère BIC associé. Cette prise en compte de la transformation appliquée aux données dans le calcul du BIC a été utilisée auparavant dans un autre domaine (Thomas et al., 2008).

5.3 Illustration sur des données simulées

5.3.1 Paramètres de simulation

Afin d'illustrer la comparaison de modèles proposée, nous simulons des données sous le modèle de mélange de lois de Poisson détaillé à l'équation (5.1), en fixant le nombre de conditions expérimentales p à 3, le nombre de gènes n à 5000, les facteurs de normalisation $(s_1, s_2, s_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ et les facteurs w_i à partir du jeu de données réelles de Mach et al. (2014) (jeu de données décrit dans la section suivante) en sélectionnant aléatoirement trois conditions expérimentales et n gènes parmi les gènes du jeu de données ayant au moins 20 comptages par gènes. Les valeurs des w_i varient ainsi de 20 à 1 800 000 comptages. Nous fixons ensuite le nombre de classes $K = 4$, les proportions de chaque classe $(p_1, p_2, p_3, p_4) = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ et les paramètres λ_{jk} pour $j = 1, 2, 3$ et $k = 1, 2, 3, 4$ tels que $\sum_j \lambda_{jk}s_j = 1$:

$$\boldsymbol{\lambda} = \begin{pmatrix} 1.5 & 1 & 0.5 & 1.5 \\ 0.5 & 1.5 & 1 & 1 \\ 1 & 0.5 & 1.5 & 0.5 \end{pmatrix}.$$

5.3.2 Résultats

La figure 5.1 (gauche) illustre les comptages simulées transformés pour la condition 1 versus la condition 2. On constate que l'hypothèse de modélisation des données transformées par un modèle de mélange de lois gaussiennes est cohérente : les différentes composantes simulées peuvent être représentées par des ellipses.

Conformément au résultat attendu, la figure 5.1 (droite) montre que le BIC du modèle de mélange gaussien, ajusté pour la t-transformation des données (5.2) est inférieur au BIC du modèle de mélange de lois de Poisson pour un nombre de classes supérieur ou égal à 4. On remarque également que le modèle de mélange de lois gaussiennes surestime fortement le nombre de composantes du mélange et sélectionne 14 classes au lieu des 4 classes simulées.

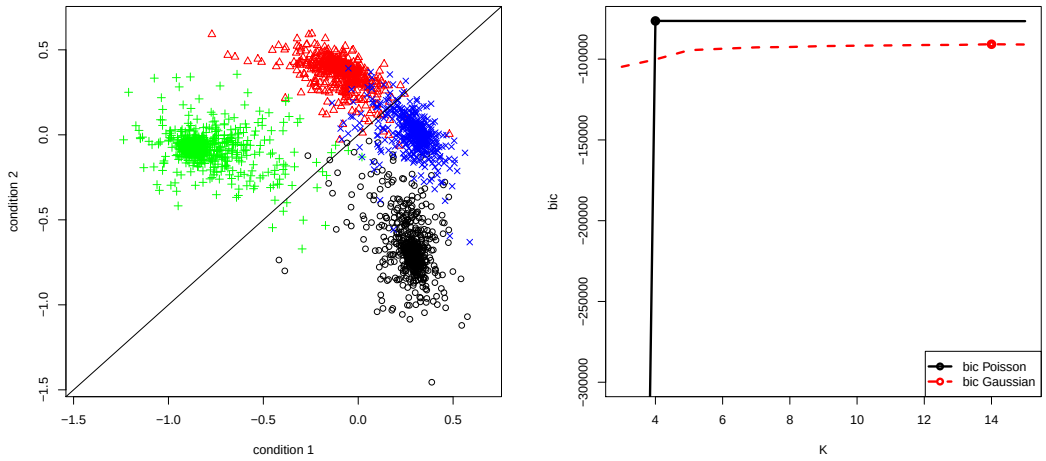

FIGURE 5.1 – *A gauche*, comptages simulés sous un modèle de mélange de lois de Poisson et t-transformés pour la conditions 1 versus la condition 2. Les différentes couleurs et symboles correspondent aux quatre classes simulées. *A droite,* BIC du modèle de mélange de Poisson et BIC du modèle de mélange gaussien ajusté pour la transformation de données t pour un nombre de classes variant de 3 à 15. Le point sur chaque courbe BIC indique le nombre de classes sélectionné par chacun des modèles.

5.4 Illustration sur des données réelles

Pour plusieurs jeux de données RNA-seq, nous effectuons la classification des gènes à l'aide du modèle de mélange de Poisson sur les données de comptage brutes, et à l'aide du modèle de mélange gaussien sur les données transformées.

5.4.1 Description des données

Cellules hépatiques chez l'humain : Sultan et al. (2008) ont analysé l'expression des gènes dans les cellules humaines embryonnaires du rein (HEK293T) et dans les cellules de la lignée Ramos B en effectuant le séquençage de deux réplicats biologiques dans chaque type de cellule par la technologie RNA-seq. Après avoir supprimé les gènes peu exprimés, nous effectuons la classification des 4959 gènes restants.

Evolution embryonnaire chez la drosophile : Dans le cadre du projet modENCODE visant à completer l'annotation fonctionnel du génome de la mouche *Drosophila melanogaster*, Graveley et al. (2011) ont analysé les différences d'expression géniques à différents stades du développement de la mouche. On s'intéresse à 12 cellules d'embryons collectées toutes les deux heures pendant 24 heures. Ces 12 réplicats biologiques sont séquencés à l'aide de la technologie RNA-seq. Le tableau de comptage des données a été extrait de la base de données ReCount (Frazee et al., 2011). Les 13164 gènes exprimés sont classés.

Tissus de l'intestin grêle chez le porc : Mach et al. (2014) ont analysé les différences d'expression entre trois tissus (le duodenum, le jejunum et l'ileum) de l'intestin grêle de quatre porcelets sains. Après avoir sélectionné les gènes différentiellement exprimés entre ces trois tissus à l'aide d'un modèle linéaire généralisé basé sur une loi négative binomiale (Robinson and Oshlack, 2010), on effectue la classification des 4021 gènes restants.

5.4.2 Résultats

Sur les Figures 5.2, nous constatons que deux des trois jeux de données de Graveley et Mach sont mieux ajustés par le modèle de mélange de lois gaussiennes sur les données transformées que par le modèle de Poisson. Le mélange de Poisson s'ajuste mieux au jeu de données de Sultan. Cependant, la différence entre les valeurs des deux BIC n'est pas comparable entre les différents jeux de données : sur les données de Sultan, les valeurs des BIC pour le mélange de Poisson et le mélange gaussien sont relativement proches alors que le BIC du mélange gaussien est très supérieur au BIC du mélange de Poisson sur les données de Graveley et Mach. On constate également que le choix de modèle réalisé par le critère BIC est plus parcimonieux pour le modèle ajustant le mieux les données : le BIC du mélange de Poisson sélectionne 27 classes sur les données Sultan alors que le BIC du mélange gaussien sélectionne plus de 30 classes. Sur les données Graveley et Mach, le critère BIC du mélange de Poisson ne parvient à sélectionner un modèle (ce BIC augmente toujours, même pour des très grandes valeurs de K), alors que le BIC du mélange gaussien sélectionne moins de 50 classes pour les deux jeux de données.

La technologie RNA-seq évolue et les jeux de données générés au moment de l'émergence de la technologie ne ressemblent pas forcement aux jeux de données plus récents. Les plans d'expérience sont de plus en plus complexes et le nombre de réplicats biologiques à disposition augmente. Le modèle de mélange de lois gaussiennes semble mieux

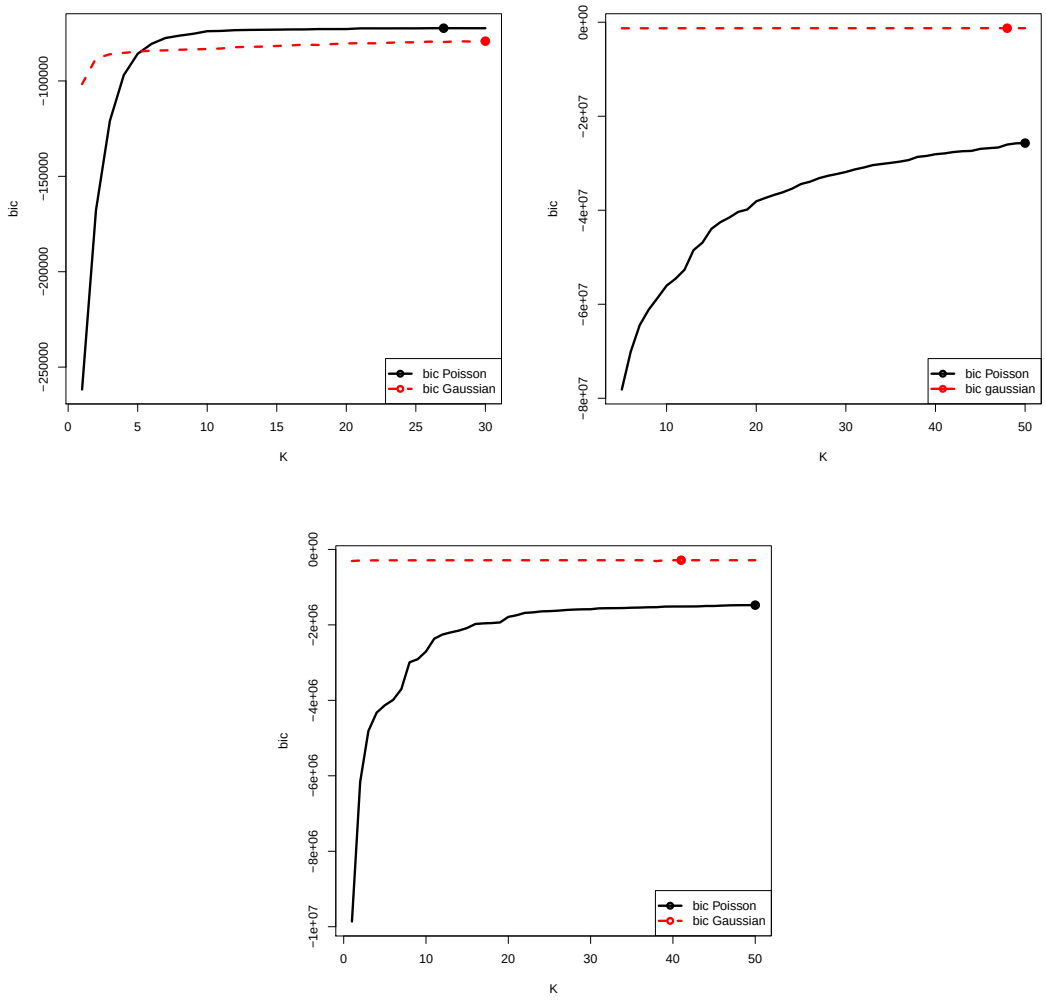

FIGURE 5.2 – BIC du modèle de mélange de Poisson sur données brutes et du modèle de mélange gaussien sur données transformées pour les données de Sultan (haut, gauche), de Graveley (haut, droite) et de Mach (bas).

prendre en compte ces plans d'expérience complexes. Cependant, une généralisation de ce résultat nécessiterait de comparer plus de trois jeux de données.

5.5 Conclusion

À l'aide du critère BIC, on peut déterminer si une transformation des données fournit un meilleur ajustement des données et comparer différentes stratégies de modélisation.

Cette comparaison est pertinente pour la classification des données RNA-seq par modèle de mélange, pour lesquels deux alternatives existent : modélisation des données à l'aide de modèle de mélange discret ou à l'aide de modèle de mélange de lois gaussiennes. Nous avons constaté qu'il n'existe pas de modèle le mieux adapté aux données RNA-seq. La comparaison de critère BIC permet ainsi de sélectionner le modèle le plus adapté aux données. Sur les trois jeux de données étudiés, la transformation logarithmique des données que nous avons proposée fournit souvent des modèles beaucoup plus convaincants que le modèle de mélange de Poisson.

Dans ce chapitre, nous avons utilisé le critère BIC pour comparer les différentes stratégies. On peut cependant utiliser d'autres critères basés sur la log-vraisemblance pénalisée des données, comme le critère ICL ou le critère SICL .

Chapitre 6

A model selection criterion for model-based clustering of annotated gene expression data

Résumé. En analyse de co-expression des données de gènes, on souhaite souvent interpréter les modules détectés à l'aide d'informations externes, telles que la liste potentiellement incomplète des propriétés fonctionnelles attribuées à un ensemble de gènes. Dans le cadre des modèles de mélange, nous proposons un critère de sélection de modèle prenant en compte ces informations externes fournissant ainsi un outil pertinent pour sélectionner un modèle et un nombre de classes. Ce critère *Integrated Completed Annotated Likelihood* (ICAL) est défini en ajoutant un terme d'entropie à la vraisemblance pénalisée du modèle afin de mesurer la concordance entre la partition inférée et les annotations externes. Le critère ICAL conduit à un choix de modèle plus facilement interprétable au regard des annotations fonctionnelles des gènes déjà répertoriées. On illustre l'intérêt de ce critère de sélection de modèle pour des mélanges gaussiens sur des données simulées et sur un jeu de données réel d'expression de gènes RNA-seq.

Abstract. In co-expression analyses of gene expression data, it is often of interest to interpret clusters of co-expressed genes with respect to a set of external information, such as a potentially incomplete list of functional properties for which a subset of genes may be annotated. Based on the framework of finite mixture models, we propose a model selection criterion that takes into account such external gene annotations, providing an efficient tool for selecting a relevant number of clusters and clustering model. This criterion, called the Integrated Completed Annotated Likelihood (ICAL), is defined by adding an entropy term to a penalized likelihood to measure the concordance between a clustering partition and the external annotation information. The ICAL leads to the choice of a model that is more easily interpretable with respect to the known functional gene annotations. We illustrate the interest of this model selection criterion in conjunction with Gaussian mixture models on simulated data and on real gene expression RNA-seq data.

6.1 Introduction

Genome annotation broadly refers to the set of meta-data associated with the coding regions in the genome, typically including the identification of the location of each gene as well as a determination of the functions related to the gene product (e.g., protein or RNA). In particular, gene annotations correspond to known functions related to the gene product, including molecular functions, biological pathways, or the cellular location of the gene products. A variety of well-known unified databases have been constructed with known functional annotations collected from bibliographic sources across species, including the Gene Ontology (GO) (Ashburner et al., 2000), the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000) or the MSigDB (Molecular Signatures) databases (Liberzon et al., 2011). Although such databases contain a rich source of functional information about the genome in a large variety of species (e.g., *Arabidopsis thaliana*, human, rat, mouse, fly), our knowledge of functional annotations is often far from complete (Tipney and Hunter, 2010).

In recent years, substantial improvements in high-throughput technologies, such as microarrays (Schena et al., 1995) and more recently RNA sequencing (RNA-seq) (Mortazavi et al., 2008), have enabled the simultaneous measurement of the expression levels of tens of thousands of genes. A rich body of work is now available on the appropriate statistical analyses of such gene expression data, including the analysis of differential expression (Smyth, 2004; Anders and Huber, 2010) and co-expression analyses to identify groups of genes with similar profiles across several experimental conditions or over the course of time (Yeung et al., 2001; Rau et al., 2015). The latter is of particular interest in this work, as identifying genes that share the same dynamic patterns of expression may help identify groups of genes that are involved in similar biological processes and generate hypotheses about the functional properties of poorly characterized genes (Eisen et al., 1998; Jiang et al., 2004). Reviews and comparisons of different clustering methods for gene expression data may be found in Datta (2003).

In practice, annotation databases are often used to perform *a posteriori* validation and interpretation of co-expressed gene clusters through tests of functional enrichment (Steuer et al., 2006). Such functional annotation may instead be directly integrated into the clustering model itself. For example, Tari et al. (2009) incorporate GO annotations as prior knowledge in a fuzzy c-means clustering. Verbanck et al. (2013) proposed a clustering approach based on a distance defined conjointly on the similarity among expression profiles and that among functional profiles. Pan (2006) and Huang et al. (2006) proposed including gene annotation as prior information in a stratified mixture model. However, the inclusion of gene annotation directly in the model itself in this way may be questionable, particularly when they are also used to validate the gene clusters *a posteriori*. Moreover, as gene annotations tend to be incomplete, biases may be introduced if they are directly incorporated in the model, as unannotated genes (which represent those known to be unassociated with a given function as well as those of unknown function) may be erroneously separated from annotated genes.

One alternative to such approaches is to define a clustering model that accounts for external gene annotations without directly including them in the model itself. To this end model-based clustering provides a convenient framework, as it 1) allows for a large set of clustering models to be fit to the gene expression alone, and 2) facilitates the

choice among this set a parsimonious model that simultaneously provides a good fit to the data and coherence with the external gene annotations. In this work, we address these points by proposing a model selection criterion that accounts for external gene annotations.

The rest of this paper is organized as follows. In Section 6.2, we present the context of model-based clustering and review classic model selection criteria. Our proposed annotated model selection criterion is presented in Section 6.3, and numerical illustrations of its behavior are presented on simulated data in Section 6.4 using Gaussian mixture models. Finally, we illustrate a co-expression analysis of real RNA-seq data in Section 6.5, and a discussion ends the paper.

6.2 Model-based clustering and model selection

Let $\mathbf{y}$ be the $(n \times p)$ matrix of observed gene expression, where n is the number of genes and p the number of biological samples. The vector $\mathbf{y}_i$ denotes the expression of gene i $(i = 1, \ldots, n)$ across the p samples. In the context of model-based clustering, the data $\mathbf{y}$ are assumed to be sampled from a finite mixture density of K random variables, each with parameterized density $\phi(\mathbf{y}_i; \mathbf{a}_k)$, $k = 1, \ldots, K$, where the mixture parameters $(\mathbf{a}_1, \ldots, \mathbf{a}_K)$ are all assumed to be distinct. The density of $\mathbf{y}$ may thus be written as

$$f(\mathbf{y}; K, \theta_K) = \prod_{i=1}^{n} \sum_{k=1}^{K} p_k \phi(\mathbf{y}_i; \mathbf{a}_k), \tag{6.1}$$

where $\theta_K = (p_1, \ldots, p_{K-1}, \mathbf{a}_1, \ldots, \mathbf{a}_K)$ are the parameters of the mixture model, and $(p_1, \ldots, p_K)$ are the mixing proportions with $p_k \in (0, 1)$ for all k, $\sum_{k=1}^{K} p_k = 1$.

For parameter estimation, the mixture model in Equation (6.1) may be thought of as an incomplete data structure model where $\mathbf{z}$ is the $(n \times K)$ matrix of unknown mixture labels, where $z_{ik} = 1$ if gene i is from group k and 0 otherwise. Note that this matrix defines a partition of the genes.

Using the mixture labels z, the completed density of $\mathbf{y}$ may be written as follows :

$$f(\mathbf{y}, \mathbf{z}; K, \theta_K) = \prod_{i=1}^{n} \prod_{k=1}^{K} (p_k \phi(\mathbf{y}_i; \mathbf{a}_k))^{z_{ik}}. \tag{6.2}$$

The maximum likelihood estimate $\hat{\theta}_K$ of the mixture parameters is computed through the Expectation-Maximization algorithm (Dempster et al., 1977) by replacing the unknown labels $\mathbf{z}$ in Equation (6.2) with $\hat{\mathbf{z}}$, defined as :

$$\hat{z}_{ik} = \begin{cases} 1 & \text{if } \arg\max_\ell \tau_{i\ell}(\hat{\theta}_K) = k \\ 0 & \text{otherwise,} \end{cases}$$

where $\tau_{i\ell}(\hat{\theta}_K)$ denotes the conditional probability given $\mathbf{y}_i$ of the ℓth mixture component under $\hat{\theta}_K$:

$$\tau_{i\ell}(\hat{\theta}_K) = \frac{\hat{p}_\ell \phi(\mathbf{y}_i; \hat{\mathbf{a}}_\ell)}{\sum_{t=1}^{K} \hat{p}_t \phi(\mathbf{y}_i; \hat{\mathbf{a}}_t)}.$$

In the context of model-based clustering, one important task is the choice of an appropriate model, most notably the revelant number of clusters K. To this end, a standard model selection criterion is the Bayesian Information Criterion (BIC) (Schwarz, 1978) :

$$\mathrm{BIC}(K) = \log f(\mathbf{y}; \hat{K}, \hat{\theta}_K) - \frac{\nu_K}{2}\log(n),$$

where $\hat{\theta}_K$ is the maximum likelihood estimator of the mixture parameters and ν_K the number of free parameters in the model with K components. This criterion is an asymptotic approximation of the logarithm of the integrated likelihood :

$$f(\mathbf{y}; K) = \int_{\theta_K} f(\mathbf{y}; K, \theta_K)\pi(\theta_K)d\theta_K,$$

where $\pi(\theta_K)$ is a weakly informative prior distribution on θ_K.

An alternative to the BIC is the Integrated Completed Likelihood (ICL) criterion (Biernacki et al., 2000) :

$$\mathrm{ICL}(K) = \mathrm{BIC}(K) - \mathrm{Ent}(K), \tag{6.3}$$

where $\mathrm{Ent}(K)$ is the estimated mean clustering entropy

$$\mathrm{Ent}(K) = -\sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}(\hat{\theta}_K)\log\tau_{ik}(\hat{\theta}_K) \geq 0. \tag{6.4}$$

Note that the ICL is a BIC-like approximation of the logarithm of the completed integrated likelihood :

$$f(\mathbf{y}, \mathbf{z}; K) = \int_{\theta_K} f(\mathbf{y}, \mathbf{z}; K, \theta_K)\pi(\theta_K)d\theta_K.$$

Because of the additional entropy term defined in Equation (6.4), the ICL favors models that lead to data partitions with the greatest evidence in terms of classification.

More recently, Baudry et al. (2014) proposed an ICL-like criterion that takes advantage of the potential explicative ability of external categorical variables $\mathbf{b} = (\mathbf{b}^1, \ldots, \mathbf{b}^R)$ where $u^r_{i\ell} = 1$ indicates that the gene i is in category ℓ for the r^{th} external categorical variable and 0 otherwise. The idea is to choose a classification $\mathbf{z}$ based on $\mathbf{y}$ that is coherent with $\mathbf{b}$. Assuming that $\mathbf{y}$ and $\mathbf{b}$ are conditionally independent given $\mathbf{z}$, the Supervised Integrated Completed Likelihood (SICL) criterion is an asymptotic approximation of the logarithm of the integrated completed likelihood :

$$f(\mathbf{y}, \mathbf{b}, \mathbf{z}; K) = \int f(\mathbf{y}, \mathbf{b}, \mathbf{z}; K, \theta_K)\pi(\theta_K)d\theta_K.$$

The SICL criterion is defined as follows :

$$\mathrm{SICL}(K) = \mathrm{ICL}(K) + \sum_{r=1}^{R}\sum_{\ell=1}^{U_r}\sum_{k=1}^{K} n^r_{k\ell}\log\frac{n^r_{k\ell}}{n_{k\cdot}}, \tag{6.5}$$

where U_r is the number of levels of the variable $\mathbf{b}^r$,

$$n^r_{k\ell} = \mathrm{card}\big\{i : z_{ik} = 1 \text{ and } u^r_{i\ell} = 1\big\},$$

and $n_{k\cdot} = \sum_{\ell=1}^{U_r} n^r_{k\ell}$. The last additional term in Equation (6.5) quantifies the strength of the link between the categorical variables $\mathbf{b}$ and the classification $\mathbf{z}$.

6.3 Taking genome annotations into account

As previously stated, the objective of this work is to make use of external gene annotations to choose a model for which clusters may be meaningfully interpreted both with respect to their expression profiles and their functional properties. To do so, we propose a novel model selection criterion that highlights the association between the clusters of expression profiles and the functional annotations associated with a subset of genes. Since gene annotations are binary variables (i.e., a gene is either annotated or unannotated), it may seem natural to directly use the SICL defined in Equation (6.5). However, in contrast to the situation considered by Baudry et al. (2014), gene annotation information is often incomplete. More precisely, for each of the G annotation terms, indexed by g, the available information $\mathbf{u}^g$ is as follows :

$$u_i^g = \begin{cases} 1 & \text{if gene } i \text{ is known to be implicated in function } g, \\ 0 & \text{if gene } i \text{ is not known to be implicated in function } g. \end{cases}$$

Note that $u_i^g = 0$ can indicate that information is missing (i.e., gene i has not yet been identified for annotation g) or that gene i is known to be unrelated to annotation g. As such, $u_i^g = 0$ does not represent the null level of variable and thus represents an incomplete binary variable. For this reason, the SICL criterion is not an appropriate measure of the link between an external annotation $\mathbf{u}^g$ and a classification $\mathbf{z}$, and a specific criterion must be defined to incorporate the gene annotation information into the model selection step. To this end, we propose the Integrated Completed Annotated Likelihood (ICAL) criterion as follows.

For each gene annotation $\mathbf{u}^g$, we first define the random matrix $\mathbf{b}^g$ of latent variables indicating the allocation of the annotations among the K clusters :

$$b_{ik}^g = \begin{cases} 1 & \text{with probability } p_k^g \text{ if } u_i^g = 1, \\ 0 & \text{if } u_i^g = 0. \end{cases} \tag{6.6}$$

Each row of the matrix $\mathbf{b}^g$ is a random vector following a multinomial distribution with parameters u_i^g and $(p_1^g, \ldots, p_K^g)$ if $u_i^g > 0$, and is the null vector $\mathbf{0}$ if $u_i^g = 0$.

For the sake of simplicity, we first derive ICAL when a single external annotation $\mathbf{b}^1$ is available. ICAL aims to select the clustering model that maximises the logarithm of the integrated annotated likelihood :

$$f(\mathbf{y}, \mathbf{z}, \mathbf{b}^1; K) = \int_{\theta_K} f(\mathbf{y}, \mathbf{z}, \mathbf{b}^1; K, \theta_K) \pi(\theta_K) d\theta_K. \tag{6.7}$$

As for the definition of the SICL, the variables $\mathbf{y}$ and $\mathbf{b}^1$ are assumed to be conditionally independent given $\mathbf{z}$. Using Bayes formula, we have

$$f(\mathbf{y}, \mathbf{z}, \mathbf{b}^1; K, \theta_K) = f(\mathbf{y}, \mathbf{z}; K, \theta_K) f(\mathbf{b}^1 \mid \mathbf{y}, \mathbf{z}; K, \theta_K).$$

Note that since $\mathbf{y}$ and $\mathbf{b}^1$ are assumed to be independent given $\mathbf{z}$, the conditional distribution of $\mathbf{b}^1$ given $\mathbf{z}$ does not depend on $\mathbf{y}$ or the mixture parameters. Thus, as $f(\mathbf{b}^1 \mid \mathbf{y}, \mathbf{z}; K, \theta_K) = f(\mathbf{b}^1 \mid \mathbf{z}; K)$, it follows that :

$$\log f(\mathbf{y}, \mathbf{z}, \mathbf{b}^1; K) = \log f(\mathbf{b}^1 \mid \mathbf{z}; K) + \log \int_{\theta_K} f(\mathbf{y}, \mathbf{z}; K, \theta_K) \pi(\theta_K) d\theta_K. \tag{6.8}$$

The last term in Equation (6.8) can be approximated with ICL(K) from Equation (6.3), and the first term may be approximated with

$$\log f(\mathbf{b}^1 \mid \hat{\mathbf{z}}; K) = \sum_{k=1}^{K} n_k^1 \log \frac{n_k^1}{n^1},$$

where $n^1 = \text{card}\{i : u_i^1 = 1\}$ and $n_k^1 = \text{card}\{i : \hat{z}_{ik} = 1 \text{ and } u_i^1 = 1\}$. Finally, an asymptotic approximation of the expression in (6.7) leads to the Integrated Completed Annotated Likelihood (ICAL) criterion :

$$\text{ICAL}(K) = \text{ICL}(K) + \sum_{k=1}^{K} n_k^1 \log \frac{n_k^1}{n^1}.$$

The generalization of this criterion to the case where $G > 1$ gene annotations are available is straightforward. The aim is now to maximize the logarithm of the integrated annotated likelihood :

$$\log f(\mathbf{y}, \mathbf{z}, \mathbf{b}^1, \ldots, \mathbf{b}^G; K) = \log \int_{\theta_K} f(\mathbf{y}, \mathbf{z}, \mathbf{b}^1, \ldots, \mathbf{b}^G; K, \theta_K) \pi(\theta_K) d\theta_K.$$

Assuming that $\mathbf{b}^1, \ldots, \mathbf{b}^G$ and $\mathbf{y}$ are conditionally independent given $\mathbf{z}$, we have

$$\log f(\mathbf{y}, \mathbf{z}, \mathbf{b}^1, \ldots, \mathbf{b}^G; K) = \log f(\mathbf{b}^1, \ldots, \mathbf{b}^G; \mathbf{z}, K) + \log \int_{\theta_K} f(\mathbf{y}, \mathbf{z}; K, \theta_K) \pi(\theta_K) d\theta_K.$$

Assuming in addition that $\mathbf{b}^1, \ldots, \mathbf{b}^G$ are independent and that gene annotations are missing at random, we can write

$$f(\mathbf{b}^1, \ldots, \mathbf{b}^G; \mathbf{z}, K) = \prod_{g=1}^{G} f(\mathbf{b}^g | \mathbf{z}, K), \tag{6.9}$$

leading to the generalized ICAL criterion :

$$\text{ICAL}(K) = \text{ICL}(K) + \sum_{g=1}^{G} \sum_{k=1}^{K} n_k^g \log \frac{n_k^g}{n^g}. \tag{6.10}$$

Comparing ICAL and SICL If we ignore the uncertainty associated with $u_i^g = 0$ (i.e., that gene i could either be unassociated with function g or that this information is missing), the SICL criterion could be considered to choose the model dimension K. In this case, using the notation from Section 6.2 and defining n_k the size of the cluster k, the SICL may be written as follows :

$$\text{SICL}(K) = \text{ICL}(K) + \text{pen}_{\text{SICL}},$$

where

$$\begin{aligned}\text{pen}_{\text{SICL}} &= \sum_{g=1}^{G}\sum_{k=1}^{K} n_{k1}^{g} \log \frac{n_{k1}^{g}}{n_{k.}^{g}} + \sum_{g=1}^{G}\sum_{k=1}^{K} n_{k0}^{g} \log \frac{n_{k0}^{g}}{n_{k.}^{g}}, \\ &= \sum_{g=1}^{G}\sum_{k=1}^{K} n_{k1}^{g} \log n_{k1}^{g} + \sum_{g=1}^{G}\sum_{k=1}^{K} n_{k0}^{g} \log n_{k0}^{g} - G\sum_{k=1}^{K} n_{k} \log n_{k}.\end{aligned}$$

On the other hand, using the notation from Section 6.2 and defining $n_{.1}^{g} = \sum_{k=1}^{K} n_{k1}^{g}$, the ICAL may be written as follows :

$$\text{ICAL}(K) = \text{ICL}(K) + \text{pen}_{\text{ICAL}},$$

where

$$\begin{aligned}\text{pen}_{\text{ICAL}} &= \sum_{g=1}^{G}\sum_{k=1}^{K} n_{k1}^{g} \log \frac{n_{k1}^{g}}{n_{.1}^{g}}, \\ &= \sum_{g=1}^{G}\sum_{k=1}^{K} n_{k1}^{g} \log n_{k1}^{g} - \sum_{g=1}^{G} n_{.1}^{g} \log n_{.1}^{g}.\end{aligned}$$

We note that the last term in the equation above is a constant independent of K. Finally, we can rewrite ICAL as a function of SICL :

$$\text{ICAL}(K) = \text{SICL}(K) - \sum_{g=1}^{G}\sum_{k=1}^{K} n_{k0}^{g} \log n_{k0}^{g} + G\sum_{k=1}^{K} n_{k} \log n_{k} + \text{constant}. \tag{6.11}$$

From Equation (6.11), we note that the SICL takes into account both modalities (0 and 1) of the external variables $\mathbf{u}$, while the ICAL discards the null modality (the $-\sum_{g=1}^{G}\sum_{k=1}^{K} n_{k0}^{g} \log n_{k0}^{g}$ term). Moreover, it can be seen that the ICAL penalises a large number of clusters, while the SICL does not (the $G\sum_{k=1}^{K} n_{k} \log n_{k}$ term). As such, the ICAL tends to select parsimonious models with a relatively small number of clusters, as compared to SICL.

It is also helpful to consider the behavior of the ICAL and SICL criteria in extreme conditions. If the number of clusters K equals 1, the ICAL penalty pen_{ICAL} equals zero whereas SICL penalty pen_{SICL} is not null ($\sum_{g=1}^{G} n_{1}^{g} \log \frac{n_{1}^{g}}{n} + \sum_{g=1}^{G} n_{0}^{g} \log \frac{n_{0}^{g}}{n}$). In contrast, if the number of clusters K is equal to the number of observations, with one gene per cluster, the SICL penalty pen_{SICL} equals zero whereas the ICAL penalty is not null ($\sum_{g=1}^{G} n_{1}^{g} \log n_{1}^{g}$). In general, ICAL tends to merge clusters to group genes annotated for the same function, reducing the number of optimal clusters K with respect to the optimal number of clusters selected by ICL. SICL tends to split clusters in order to obtain clusters made up only of annotated genes, increasing the number of optimal clusters with respect to the optimal number of clusters selected by ICL. In other words, SICL tends to select more complex models than ICL while ICAL tends to favor more parsimonious models than ICL. Note that this behavior of ICAL and SICL is a general trend, not a rule : ICAL does not always merge clusters and SICL does not always split

them since clusters for different solutions are not necessarily nested in each other.

Code to implement our method is available in the R package ICAL, which may be found at the following website : https ://github.com/Gallopin/ICAL.

6.4 Numerical illustrations

6.4.1 Simulation settings

To illustrate the behavior of the proposed ICAL criterion, we consider a numerical example. We simulate 200 observations from a mixture of four bivariate Gaussian distributions, 100 independent times (see parameters in Table 6.1). The first two components are close to one another while the third and fourth are clearly distinct from the first two and also distinct from each other. For a given model indexed by K, the estimation of parameters is performed with the R package `Rmixmod` (Biernacki et al., 2006; Lebret et al., 2013) for a Gaussian mixture model with diagonal variance matrix (that is, the $p_k L_k B_k$ model in the notation of the `Rmixmod` package, corresponding to clusters with variable proportions, variable volumes, variable shapes and vertical or horizontal orientation). We estimate the parameters for models with the number of clusters K varying from 1 to 10 and perform model selection to select the most appropriate number of clusters. Over the 100 replicated datasets, the BIC most frequently selects four clusters (81 times). Indeed, we note that these clusters correspond to the simulated Gaussian components. The ICL criterion selects either three (54 times) or four clusters (46 times), as it tends to merge the two similar components (1 and 2 from Table 6.1) .

Component	Mixing proportions	Component distribution
1	0.25	$\mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1.7\end{pmatrix}\right)$
2	0.25	$\mathcal{N}\left(\begin{pmatrix}0\\5\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1.7\end{pmatrix}\right)$
3	0.25	$\mathcal{N}\left(\begin{pmatrix}9\\8\end{pmatrix}, \begin{pmatrix}0.1 & 0\\0 & 1\end{pmatrix}\right)$
4	0.25	$\mathcal{N}\left(\begin{pmatrix}9\\0\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 0.8\end{pmatrix}\right)$

TABLE 6.1 – Parameters of simulated datasets : the first two components are close to one another while the third and fourth are clearly distinct from the first two and also distinct from each other.

We illustrate the potential utility of accounting for external gene annotations in model selection by simulating such annotations and performing model selection with the corresponding SICL and the ICAL criteria. We simulate three types of functional annotations : $\mathbf{u}_\mathrm{A}$, $\mathbf{u}_\mathrm{B}$ and $\mathbf{u}_\mathrm{C}$ (see Figure 6.4.1). The genes annotated for the first function $\mathbf{u}_\mathrm{A}$ are shared by the two closest mixture components (components 1 and 2 from Table 6.1). This annotation is designed to be *associated to the components* in the sense that it

suggests the interest of merging the two clusters, as they share similar joint distributions and external annotations. The genes annotated for the second function $\mathbf{u}_B$ are shared only by the two clearly distinct components (components 3 and 4 from Table 6.1). This annotation is designed to be *unassociated with the components* : although the components share a similar function, their joint distributions are too distinct to be merged from a modelling point of view. Finally, the genes annotated for the third function $\mathbf{u}_C$ are randomly spread over the four components : meaning the annotation is *mixed* (half associated / half unassociated). For each function, we simulate the annotation using binomial random variables, with parameters fixed to yield on average n_{annot} annotated genes over 200 possible genes. In the following, we tested two levels of annotation densities $d_{\text{annot}} = \frac{n_{\text{annot}}}{n}$: $d_{\text{annot}} = 0.05$ and $d_{\text{annot}} = 0.25$. Since the ICAL and SICL criteria can be used with more than one external annotation, we also illustrate the potential utility of including more than one annotation. For each set of annotations used in our simulation, the number of annotations varies from one to twelve. Each annotation in the set is simulated independently.

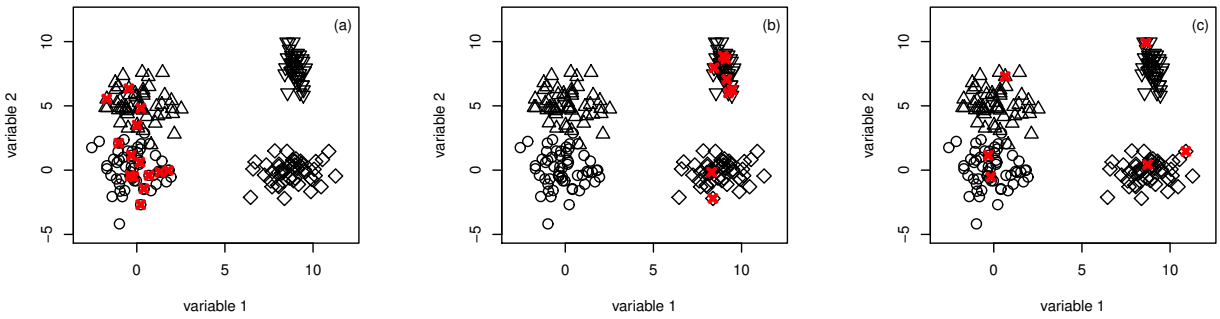

FIGURE 6.1 – Illustration of a simulated dataset and three annotation patterns. For each figure, the 200 observations are drawn from a mixture of Gaussian bivariate components whose parameters are defined in Table 6.1 : circles, triangles, inverted triangles and diamonds correspond to components 1, 2, 3 and 4. The three figures correspond to three annotation patterns : associated annotation $\mathbf{u}_A$ (a), unassociated annotation $\mathbf{u}_B$ (b) and mixed annotations $\mathbf{u}_C$ (c). For each annotation, the 20 annotated genes are represented by coloured bold crosses.

6.4.2 Simulation results

All penalized criteria (BIC, ICL, SICL and ICAL) versus the number of clusters for one simulated dataset with $d_{\text{annot}} = 0.05$ are displayed in Figure 6.2 (a). Over the 100 simulated datasets, ICAL selects three clusters 72 times, merging the two closest components 1 and 2 (Table 6.2). This three cluster solution is meaningful with respect to the information provided by $\mathbf{u}_A$, as all annotated genes are attributed to the same cluster. In this case, the external information provided by the associated annotation $\mathbf{u}_A$ reinforces the model selection. Using the same pattern as $\mathbf{u}_A$ (annotations shared by components 1 and 2 only), we simulate twelve independent associated annotations with density $d_{\text{annot}} = 0.05$. The evidence to merge clusters increases with the number of

annotations (one versus twelve annotations). The number of annotations in the set (12 annotations) is chosen so that the evidence to merge clusters was sufficiently strong : over the 100 simulated datasets, ICAL systematically selects a three cluster solution (Table 6.2). For this set of 12 external annotations, the peak of the ICAL displayed in Figure 6.2 (b) is much sharper than the peak of ICL. In contrast, SICL more frequently selects a four− or even five− cluster solution, as it leads to a preference of smaller clusters containing only annotated genes (i.e., a high specificity of annotation within each cluster). This demonstrates the utility of the ICAL criterion over the SICL, as it does not correctly take into account the specificity of gene annotation.

For unassociated annotation $\mathbf{u}_B$ with $d_{\text{annot}} = 0.05$, the behavior of the information criteria versus the number of clusters for one simulated dataset is displayed in Figure 6.3 (a). We note that the ICAL criterion behaves similarly to the ICL. Over the 100 simulated datasets, ICAL, as ICL, leads to some uncertainty as to whether a three cluster solution (53 times) or a four cluster solution (47 times) is best (Table 6.2). In this case, the annotation $\mathbf{u}_B$ is not related to the components and has no impact on the resulting clustering, even if the number of annotations in the set is increased to 12, each simulated with the same pattern as $\mathbf{u}_B$ as displayed in Figure 6.3 (b).

Finally, for the mixed annotation $\mathbf{u}_C$ with $d_{\text{annot}} = 0.05$, ICAL most frequently selects three clusters (67 times) or four clusters (33 times). Because the annotation $\mathbf{u}_C$ is mixed, there is less evidence to merge the two clusters on the left than in the case of the informative annotation $\mathbf{u}_A$. Using the three types of annotations at the same time $(\mathbf{u}_A, \mathbf{u}_B, \mathbf{u}_C)$, the ICAL criterion almost systematically selects three clusters (Table 6.2).

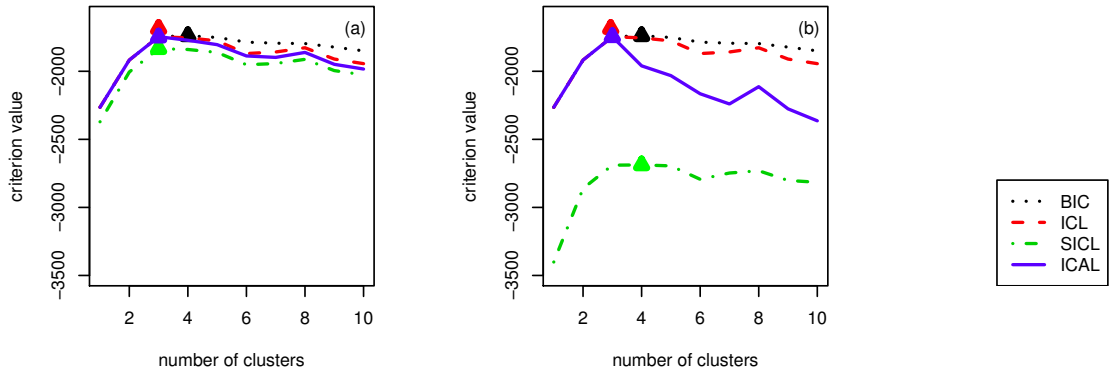

FIGURE 6.2 – BIC, ICL, SICL and ICAL information criteria versus the number of clusters on one simulated dataset for the informative annotations : $\mathbf{u}_A$ (a) and $\mathbf{u}_A^1, \ldots, \mathbf{u}_A^{12}$ (b). Triangles indicate the maximum value attained by each criterion.

The potential utility of accounting for external gene annotations in model selection is highlighted in the numerical results summarized in Table 6.2. First, these results illustrate that the SICL is not well-adapted to account for gene annotations in model selection ; at best SICL behaves like ICL and at worst, erroneously splits clusters that should be merged. However, if the external information is associated to the components, even partially so, the use of the ICAL criterion improves model selection in terms of functional interpretability. If the external information is unassociated to the components,

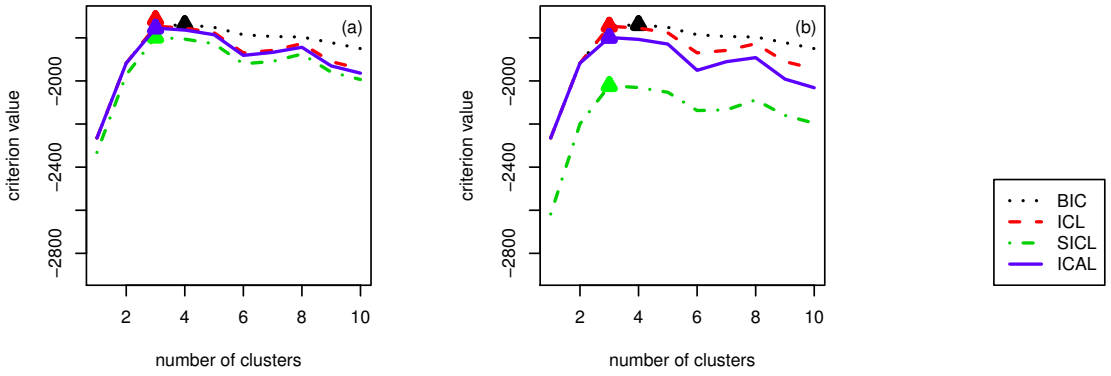

FIGURE 6.3 – BIC, ICL, SICL and ICAL information criteria versus the number of clusters on one simulated dataset for the non informative annotations : $\mathbf{u}_{\mathrm{B}}$ (a) and $\mathbf{u}_{\mathrm{B}}^{1}, \ldots, \mathbf{u}_{\mathrm{B}}^{12}$ (b). Triangles indicate the maximum value attained by each criterion.

the ICAL criterion simply behaves like the ICL.

To assess the influence of annotation density on our results, we repeated the experiment for a higher density ($d_{\mathrm{annot}} = 0.25$), as summarized in Table 6.3. Generally speaking, we note that when the number of annotations in the set increases, the evidence to merge clusters is stronger ; for a higher density of annotations, a smaller number of annotations in the set is needed to merge clusters. In particular, for a lower density of annotations ($d_{\mathrm{annot}} = 0.05$), increasing the number of associated low density annotations in the set (from one to twelve) also increases the level of confidence to merge clusters ; clusters 1 and 2 are merged 72% of the time for a single annotation, and systematically merged in all simulated datasets for a set of twelve annotations (Table 6.2). On the other hand, for a higher density of annotations ($d_{\mathrm{annot}} = 0.25$), clusters 1 and 2 are systematically merged even for a set of one single annotation (Table 6.3).

If the annotation density is high (i.e., where more than half of the genes in clusters 3 and 4 are annotated), multiple unassociated annotations can lead to overly parsimonious solutions, such as the two cluster solutions in Table 6.3.

		K	1	2	3	4	5	6	7	8	9	10
		BIC			19	81						
		ICL			53	47						
Associated annotations	$\mathbf{u}_A$	SICL			51	49						
		ICAL			72	28						
	$\mathbf{u}_A^1, \ldots, \mathbf{u}_A^{12}$	SICL			27	70	3					
		ICAL			100							
Unassociated annotations	$\mathbf{u}_B$	SICL			53	46	1					
		ICAL			53	47						
	$\mathbf{u}_B^1, \ldots, \mathbf{u}_B^{12}$	SICL			53	44	3					
		ICAL			53	47						
Mixed annotations	$\mathbf{u}_C$	SICL			50	50						
		ICAL			67	33						
Multiple annotations	$\mathbf{u}_A, \mathbf{u}_B, \mathbf{u}_C$	SICL			48	51	1					
		ICAL			81	19						

TABLE 6.2 – Number of simulated datasets for which each model ($K = 1, \ldots, 10$) was selected by BIC, ICL, SICL and ICAL for several external annotations with density $d_{\text{annot}} = 0.05$ over 100 independent datasets simulated with parameters detailed in Table 6.1. The model most commonly selected for each criterion is highlighted in red.

		K	1	2	3	4	5	6	7	8	9	10
		BIC			19	81						
		ICL			53	47						
Associated annotations	$\mathbf{u}_A$	SICL			52	48						
		ICAL			100							
	$\mathbf{u}_A^1, \mathbf{u}_A^2, \mathbf{u}_A^3$	SICL			53	47						
		ICAL			100							
Unassociated annotations	$\mathbf{u}_B$	SICL			53	47						
		ICAL			53	47						
	$\mathbf{u}_B^1, \mathbf{u}_B^2, \mathbf{u}_B^3$	SICL			53	47						
		ICAL		6	47	47						
Mixed annotations	$\mathbf{u}_C$	SICL			52	48						
		ICAL			94	6						
Multiple annotations	$\mathbf{u}_A, \mathbf{u}_B, \mathbf{u}_C$	SICL			51	49						
		ICAL			100							

TABLE 6.3 – Number of simulated datasets for which each model ($K = 1, \ldots, 10$) was selected by BIC, ICL, SICL and ICAL for several external annotations with density $d_{\text{annot}} = 0.25$ over 100 independent datasets simulated with parameters detailed in Table 6.1. The model most commonly selected for each criterion is highlighted in red.

6.5 RNA-seq data analysis

6.5.1 Presentation of the RNA-seq data and clustering settings

Mach et al. (2014) analyzed transcriptome differences in the small intestine of healthy piglets to better understand their immune response. The expression of 24924 genes across 12 samples was measured using RNA-seq, corresponding to 3 different tissues (the duodenum, the jejunum and the ileum), each sequenced for 4 different healthy piglets. The raw data are available at NCBI's SRA repository (PRJNA221286 BioProject ; accessions SRR1006118 to SRR1006133), and sequencing reads were pre-processed (i.e., quality control, alignment, and esimation of gene expression) as described in Mach et al. (2014). We performed a differential analysis using a negative binomial generalized linear model as implemented in the `edgeR` package version 3.4.2 (Robinson and Oshlack, 2010). We identified 4021 genes as differentially expressed among any of the tissues after controlling the false discovery rate (FDR) below the level 0.05 with the approach of Benjamini and Hochberg (1995). For the following co-expression analysis, we restrict our attention to this set of differentially expressed genes.

The co-expression analysis was performed on the logarithm of the counts scaled by the library size times one million. This transformation was used in Law et al. (2014) to stabilize the unequal variabilities typical of RNA-seq data and enable the use of a Gaussian linear model. The count expression y_{ij} of gene i for sample j ($i = 1, \ldots, n; j = 1, \ldots, p$) is transformed as follows : $\text{log-cpm}(y_{ij}) = \log_2\left(\frac{y_{ij}+0.5}{N_j+1} \times 10^6\right)$, where N_j is the total count normalization factor for sample j computed on the full set of genes. All replicates were included in the clustering analysis following transformation, rather than averaging over replicates within each condition.

Subsequently, Gaussian mixture models were estimated for the transformed data using the `Rmixmod` package version 2.0.2 (Biernacki et al., 2006) for a number of clusters from 1 to 50. For each model, we used a *small EM* strategy for initialization (Biernacki et al., 2003) and repeated estimation 10 times.

6.5.2 Presentation of functional annotation data

The Molecular Signatures Database (Liberzon et al., 2011) was built by the Brain Institute and provides collections of annotated gene sets for use with the Gene Set Enrichment Analysis software (Subramanian et al., 2005). The Molecular Signatures Database (MSigDB) contains collections of gene sets from several sources : positional gene sets, curated gene sets from online pathway databases, motif gene sets, computational gene sets, GO gene sets, oncogenic canonical pathways and immunologic signatures. We used the Canonical Pathways (CP) gene sets collection, compiling 1320 canonical representations of biological processes curated by domain experts from online metabolic and signaling pathways databases such as the KEGG (`http://www.genome.jp/kegg`), BioCarta (`http://www.biocarta.com`) and Reactome databases (`http://www.reactome.org`).

Among the 1320 CP in the database, 1131 are represented among the 4021 differentially expressed genes. We select the CPs for which annotated genes are overrepresented

in the set of differentially expressed genes with respect to the set of non-null genes using a Fisher's exact test. Since a test is performed for every possible annotation (i.e., each CP), we select those whose adjusted p-value is less than 0.05, after applying a Bonferroni correction for multiple testing. This procedure yields 10 CPs of interest, as described in Table 6.4.

CP	Name	DE genes	Total genes
1	Reactome metabolism of lipids and lipoproteins	141	480
2	Reactome transmembrane transport of small molecules	124	415
3	Reactome hemostasis	99	468
4	Reactome SLC mediated transmembrane transport	73	243
5	Reactome phospholipid metabolism	54	200
6	Reactome fatty acid triacylglycerol and ketone body metabolism	53	170
7	KEGG PPAR signaling pathway	34	71
8	KEGG ECM receptor interaction	34	86
9	Reactome transport of inorganic cations anions and amino acids oligopeptides	33	96
10	KEGG peroxisome	31	80

Table 6.4 – Number of genes annotated for each canonical pathway (CP) : among the 4021 differentially expressed (DE) genes and among the full CP gene set collection of the MSigDB database.

6.5.3 Model selection

We compare the results of model selection performed by the four different criteria presented in Sections 6.2 and 6.3 : BIC selects 28 clusters, ICL and SICL select 23 clusters while ICAL selects 20 (see Figure 6.4). Figures 6.5 and 6.6 are heatmaps of the resulting clusters for the ICL and ICAL solutions respectively. The approximate correspondences between clusters in the ICAL and ICL solutions are displayed in Table 6.5. Although the result of the former is not perfectly nested in the latter, in many cases the attribution of genes to clusters in the ICAL solution is a result of collapsing or partially collapsing several clusters from the ICL solution. For example, the ICAL merges most of cluster 2 and parts of clusters 5 and 18 from the ICL solution because they share similar expression profiles and functional annotations, as illustrated in Figure 6.7. This suggests that the ICL favors a slightly more complex solution, as expected ; we next investigate whether the more parsimonious solution of the ICAL appears to be coherent given the set of CP used.

For the ICL and ICAL solutions, we examine associations between clusters and CP using Fisher's exact test. Significant p-values are summarized in Table 6.6. The ICAL criterion yields a clustering that maximizes the number of genes annotated in each cluster for each CP while still only grouping genes that share sufficiently similar expression profiles. For example, we note that CP8 is associated with two different clusters in the ICL solution, while it is associated with a single cluster in the ICAL solution ; similarly, CP10 is associated with three clusters in the ICL solution and only two clusters in the ICAL solution. On the other hand, although clusters 10 and 17 in the ICAL solution both share annotations for CP10, these clusters are not collapsed

into one using the proposed criterion, as their expression dynamics are too different. As such, the ICAL solution appears to enable the identification of more biologically interpretable clusters than the ICL, while still ensuring that the clustered genes share sufficiently similar expression dynamics.

Finally, we note that the ICAL solution exhibits two clusters of particular interest with respect to the biological processes studied : Cluster 5 (379 genes) is associated with CP3 (reactome homeostasis, p=0.0002) and CP8 (KEGG ECM receptor interaction, p=0.00001). Cluster 10 (297 genes) is associated with CP1 (reactome metabolism of lipids and lipoproteins, p=0.002), CP6 (reactome fatty acid triacylglycerol and ketone body metabolism, p=0.005) and CP10 (KEGG peroxisome, p=0.0001), all of which correspond to fatty acid metabolism. Both clusters 5 and 10 contain unknown genes that may be good candidates for follow-up studies to determine whether they may be implicated in the corresponding canonical pathways.

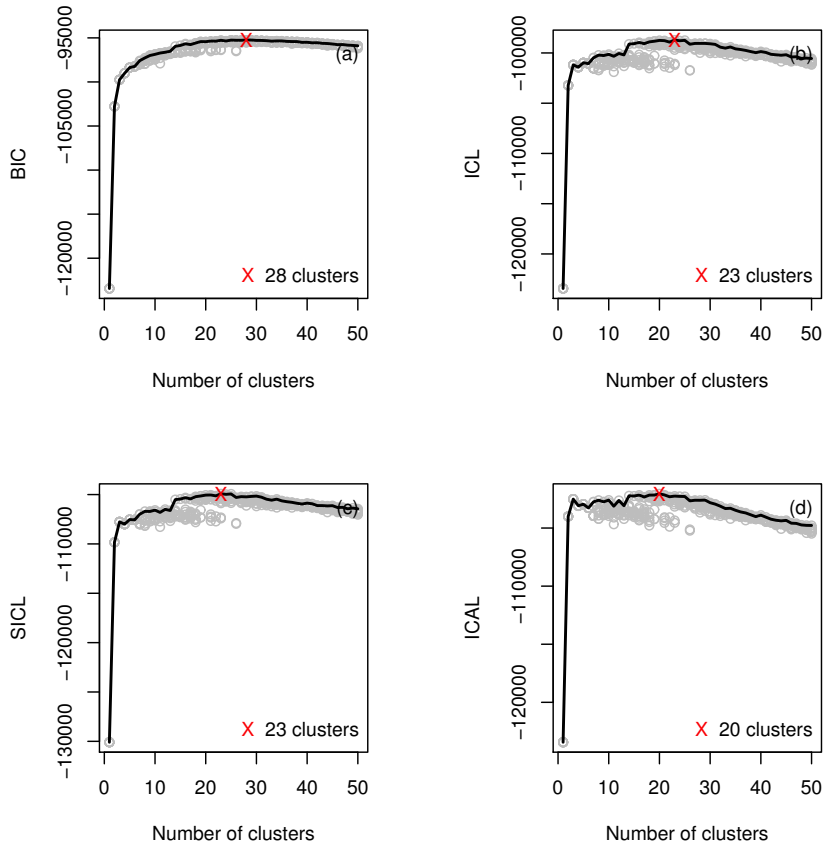

FIGURE 6.4 – BIC, ICL, SICL and ICAL information criteria (respectively a, b, c and d) versus the number of clusters for the pig RNA-seq data for 10 independent initializations, represented by 10 grey circles for each number of clusters K. Solid lines link the maximum of the criteria over the 10 initializations over the collection of models. The red crosses correspond to the maximum of the criteria, which correspond to the selected models.

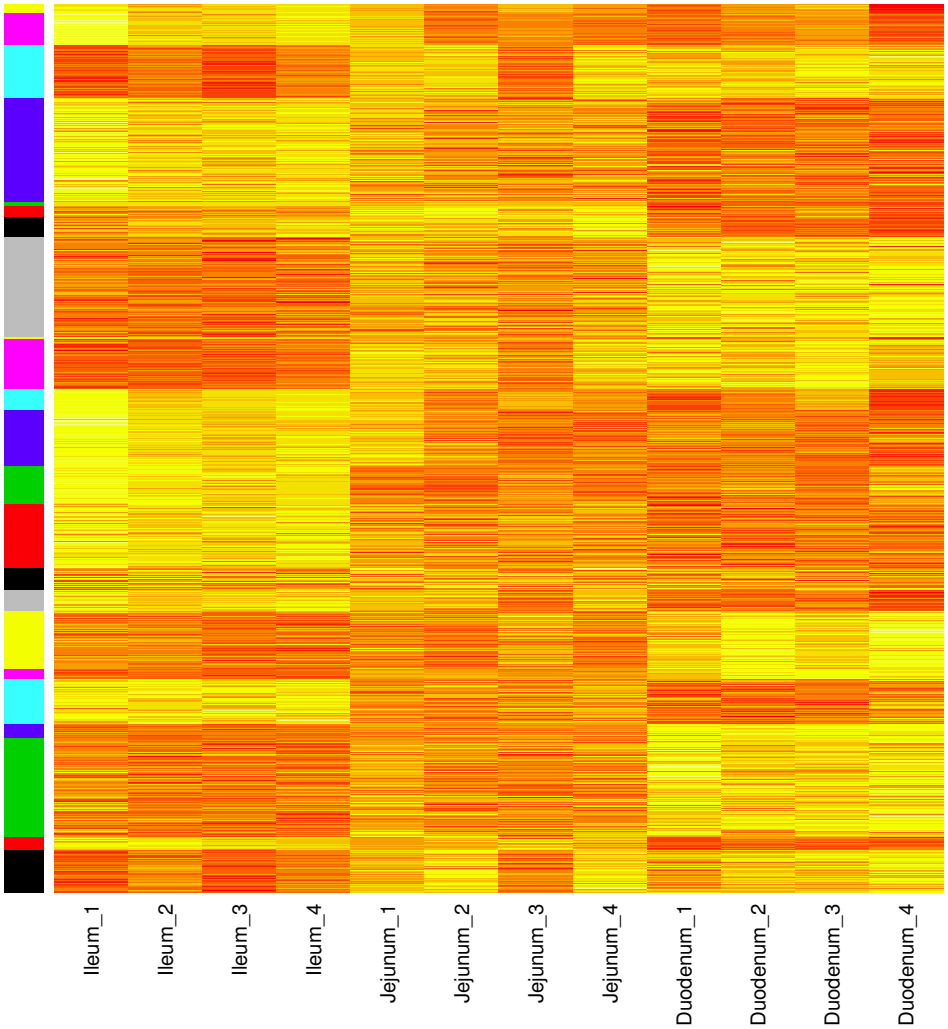

FIGURE 6.5 – Heatmap of the ICL clusters. The RNA-seq data, expressed in log-cpm is centered and scaled. Colors on the heatmap reflect the level of normalized expression : low in red, high in yellow. The 23 clusters are indicated by colors on the left side of the heat map.

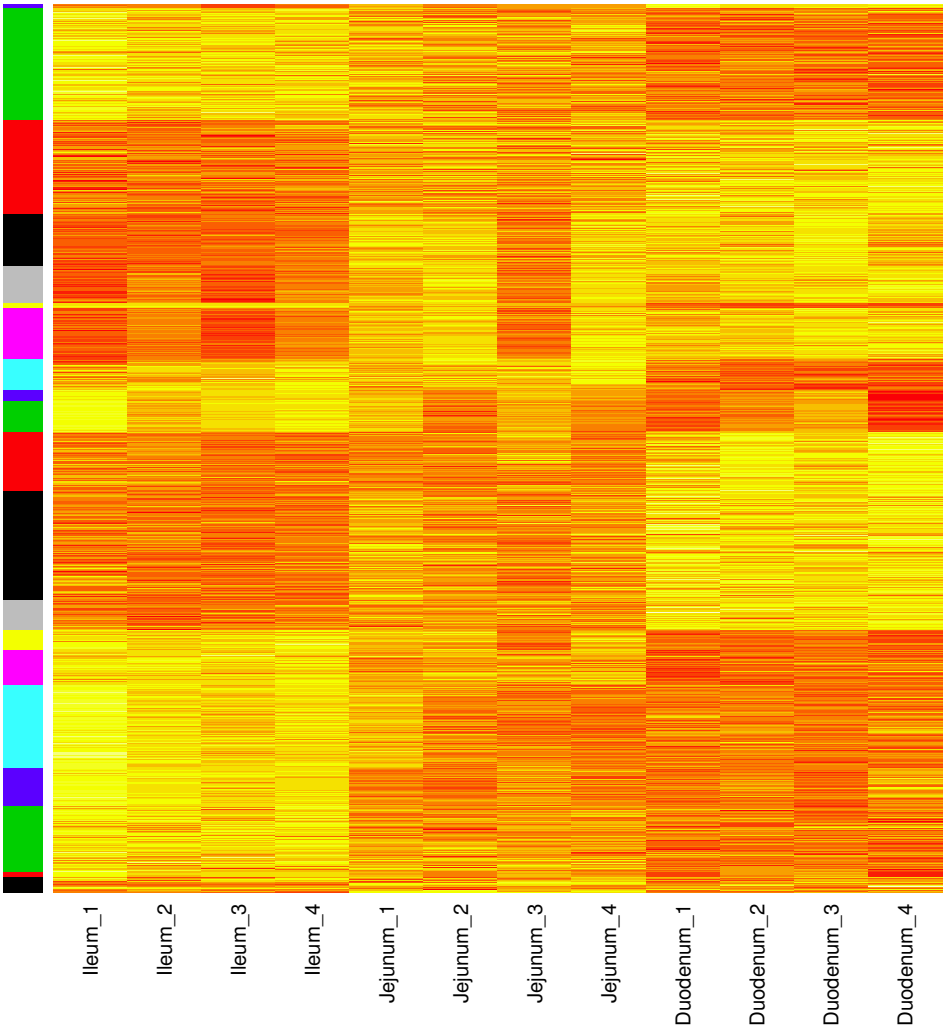

FIGURE 6.6 – Heatmap of the ICAL clusters. The RNA-seq data, expressed in log-cpm is centered and scaled. Colors on the heatmap reflect the level of normalized expression : low in red, high in yellow. The 20 clusters are indicated by colors on the left side of the heat map.

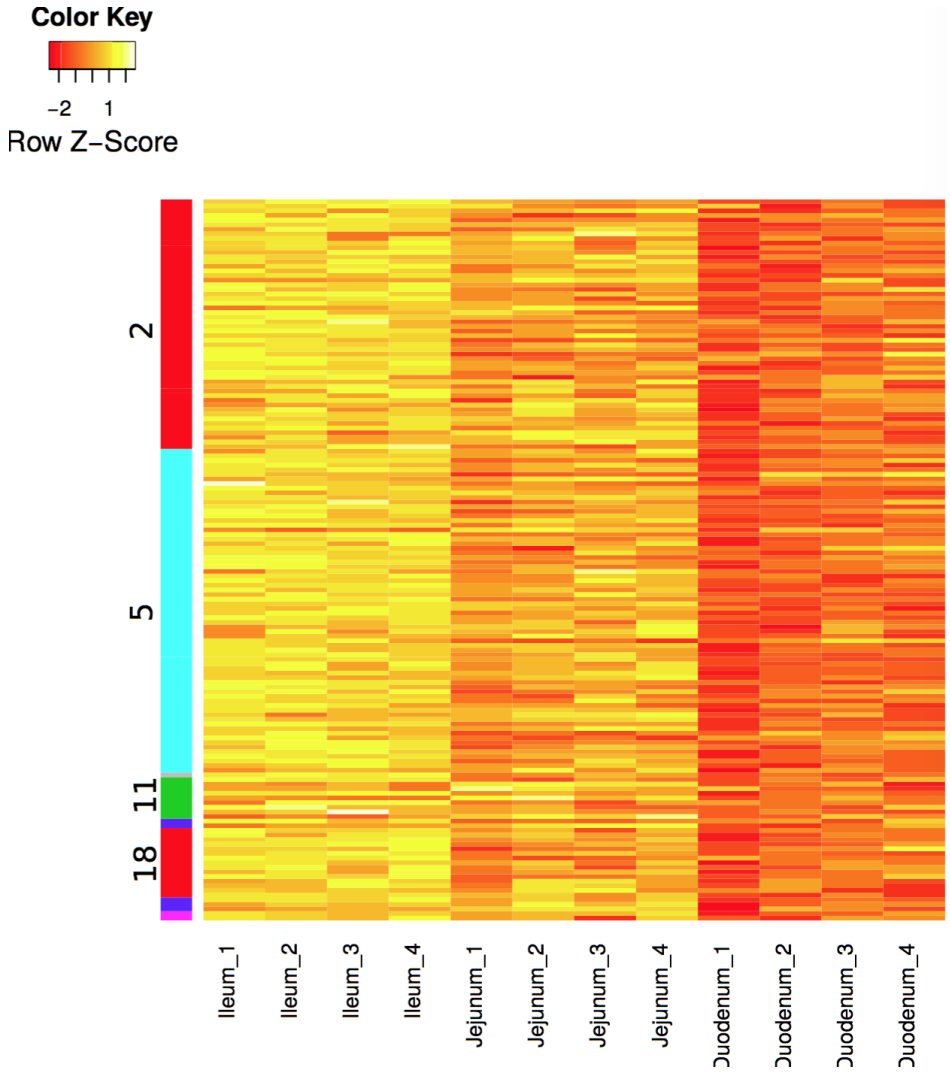

FIGURE 6.7 – Heatmap of cluster 6 of the ICAL solution. The RNA-seq data, expressed in log-cpm is centered and scaled. Colors on the heatmap reflect the level of normalized expression : low in red, high in yellow. The colors and numbers on the left side of the heatmap indicate the corresponding clusters from the ICL solution that have been merged in the ICAL solution.

ICAL clusters	ICL clusters
Cluster 1	$\frac{1}{2}$ **9**
Cluster 2	15
Cluster 3	10
Cluster 4	11
Cluster 5	$\frac{1}{2}$ **5** + **12** + $\frac{1}{2}$**20** + $\frac{1}{2}$**22**
Cluster 6	**2** + $\frac{1}{2}$**5** + $\frac{1}{3}$**18**
Cluster 7	8
Cluster 8	**4** + **9** + **16**
Cluster 9	**3** + $\frac{1}{2}$**6**
Cluster 10	**7** + $\frac{1}{2}$**6**
Cluster 11	**13** + $\frac{1}{4}$ **20** + $\frac{1}{2}$**22**
Cluster 12	23
Cluster 13	**7** + **17** + $\frac{1}{3}$ **18**
Cluster 14	21
Cluster 15	19
Cluster 16	1
Cluster 17	**14** + $\frac{1}{3}$ **18**
Cluster 18	1
Cluster 19	$\frac{1}{4}$**20** + $\frac{1}{2}$ **5**
Cluster 20	16

TABLE 6.5 – Approximate composition of the 20 clusters of the ICAL solution with respect to the 23 clusters of the ICL solution. Lines in bold correspond to clusters of the ICAL solution that are formed by several clusters or parts from clusters of the ICL solution. For example, Cluster 5 of the ICAL solution is approximately made of Clusters 12 and parts of Clusters 5, 20 and 22 of the ICL solution.

[ICL solution]

	size	CP1	CP2	CP3	CP4	CP5	CP6	CP7	CP 8	CP9	CP10
Cluster 2	**58**		⋆	⋆	⋆						
Cluster 5	**203**								⋆		
Cluster 6	**47**										⋆⋆
Cluster 7	**258**	⋆					⋆				⋆
Cluster 8	**96**					⋆⋆					
Cluster 10	**287**									⋆	
Cluster 14	**225**										⋆⋆
Cluster 22	**144**			⋆⋆					⋆⋆⋆		

[ICAL solution]

	size	CP1	CP2	CP3	CP4	CP5	CP6	CP7	CP 8	CP9	CP10
Cluster 3	**297**									⋆	
Cluster 5	**379**			⋆⋆					⋆⋆⋆		
Cluster 6	**156**		⋆⋆	⋆							
Cluster 7	**92**					⋆					
Cluster 10	**267**	⋆					⋆⋆				⋆⋆
Cluster 17	**235**										⋆⋆

TABLE 6.6 – Table of associations between clusters and CP for the ICL solution (a) and the ICAL solution (b). Associations are detected using Fisher's exact tests : the number of stars indicates the value of the p-value (⋆ below 0.01, ⋆⋆ below 0.001, ⋆⋆⋆ below 0.0001).

6.6 Discussion

In this paper, we present a novel way to incorporate functional annotations into model-based clustering of gene expression data. To this end, we have developed a model selection criterion, the Integrated Completed Annotated Likelihood (ICAL) which is designed to select the model that jointly maximizes the goodness-of-fit to the data and the association of clusters and annotations. From a biological point of view, the ICAL criterion aims to select models with more interpretable clusters than those selected by BIC or ICL. It is important to note that the functional annotations are not directly included in the clustering model and are only used to select the best model. This approach is a good compromise between two opposite strategies : including functional annotations directly in the clustering model (Morlini, 2011) or excluding them altogether and using them only to validate clusters *a posteriori*. Since we do not include annotations in the clustering model, we detect associations between annotations and clusters with a stronger evidence than if we had included the external annotations in the clustering model. In particular, the ICAL criterion is a good way to include prior biological expertise without according it too much importance, which is a good balance between what can be observed in the data and what experts expect to see in the data.

As illustrated in numerical simulations, the model selected by ICAL depends on the quality of the annotation information provided. Selecting the appropriate annotations to include in ICAL is an important step and should be done based on expert knowledge. We also suggest the use of gene annotation databases that are curated manually by experts, such as the gene sets collection from the MSigDB database (Liberzon et al., 2011). However, the choice of annotations should reflect the biological functions of interest

to a particular study, making it difficult to provide general guidelines about how such annotations should be chosen in practice. We note that if the annotations chosen are not relevant to the cluster patterns present in the data, they do not contribute any information and ICAL tends to behave like the ICL criterion.

In this work, we applied the ICAL using the framework of Gaussian mixture models, but the extension to other mixture models is straightforward ; including Poisson (Rau et al., 2015) or Dirichlet multinomial mixture models (Holmes et al., 2012). In addition, this model selection strategy may be useful for other types of data which may also be associated with incomplete external annotations (e.g., sociology, marketing).

Troisième partie

Modèle graphique pour l'inférence de réseaux

Sommaire

Chapitre 7

A hierarchical Poisson log-normal model for network inference from RNA sequencing data

Résumé. L'inférence de réseau de gènes à partir des données transcriptomiques est un défi méthodologique important et un aspect clef de la biologie des systèmes. Des méthodes ont été proposées pour l'inférence de réseau à partir des données de puces à ADN, mais ces méthodes ne sont pas directement applicables aux données RNA-seq qui sont des données discrètes et très hétérogènes. Dans ce chapitre, nous proposons un modèle hiérarchique log-normal de Poisson pénalisé afin d'inférer des réseaux de gènes sur les données RNA-seq. Basé sur des lois de Poisson, ce modèle prend en compte le caractère discret des données RNA-seq. Grâce à sa structure hiérarchique, il modèlise également la grande variabilité des données RNA-seq pour lesquelles la variance empirique est supérieure à la moyenne empirique. Nous comparons la méthode proposée à deux alternatives : un modèle graphique gaussien pénalisé appliqué sur les données log-tranformées, et un modèle graphique de Poisson log linéaire appliqué sur les données transformées par une transformation de puissance. Ces méthodes sont comparées à l'aide de simulations et à l'aide d'un jeu de données réels microRNA-seq extrait de cellules cancéreuses du sein. Sur les données simulées avec une large dispersion inter échantillons, le modèle proposé est plus performant que les autres méthodes appliquées sur données transformées en terme de sensibilité, de spécificité et d'aire sous la courbe ROC. Ces résultats montrent la nécessité de proposer des méthodes d'inférence de réseaux spécifiques aux données RNA-seq.

Abstract. Gene network inference from transcriptomic data is an important methodological challenge and a key aspect of systems biology. Although several methods have been proposed to infer networks from microarray data, there is a need for inference methods able to model RNA-seq data, which are count-based and highly variable. In this work we propose a hierarchical Poisson log-normal model with a Lasso penalty to infer gene networks from RNA-seq data ; this model has the advantage of directly modelling discrete data and accounting for inter-sample variance larger than the sample mean. Using real microRNA-seq data from breast cancer tumors and simulations, we compare this method to a regularized Gaussian graphical model on log-transformed data, and a Poisson log-linear graphical model with a Lasso penalty on power-transformed data.

For data simulated with large inter-sample dispersion, the proposed model performs better than the other methods in terms of sensitivity, specificity and area under the ROC curve. These results show the necessity of methods specifically designed for gene network inference from RNA-seq data.

7.1 Introduction

In recent years, high-throughput sequencing technology has become an essential tool for genomic studies. In particular, it allows the transcriptome to be directly sequenced (RNA sequencing), which provides count-based measures of gene expression. Typically, the first biological question arising from these data is to identify genes differently expressed across biological conditions. Because RNA-seq data are known to exhibit a large amount of variability among biological replicates, most methods for differential analysis are based either on overdispersed Poisson (Auer and Doerge, 2011) or negative binomial models (Anders and Huber, 2010; Robinson et al., 2010).

In order to study the relationships between these large numbers of genes, several authors have worked on co-expression networks and used methods based on Pearson correlation (Giorgi et al., 2013) or canonical correlation (Hong et al., 2013; Iancu et al., 2012), but no specific models have been designed for RNA-seq data. A further question is how these genes interact with each other. Inference of gene networks from transcriptomic data is indeed a key aspect of systems biology that may help unravel and better understand the underlying biological regulatory mechanisms. Various models have been proposed for network inference from microarray data, mainly based on Gaussian graphical models (Friedman et al., 2008; Meinshausen and Buhlmann, 2006). Until now, very few authors have addressed the question of network inference from RNA-seq data. Some authors simply use methods based on a Gaussian assumption for RNA-seq data (Cai et al., 2012). We propose in this paper to compare various approaches to tackle this issue.

The simplest idea is to perform an appropriate transformation of the data, using for example a Box-Cox transformation (Box and Cox, 1964) and apply methods that rely on an assumption of normality. Another possibility is to use models specifically designed for count data with large variability. Allen and Liu (Allen and Liu, 2012) recently proposed a Poisson log-linear graphical model adapted to count data. This model requires a power transformation of the data (Li et al., 2012) when the inter-sample variance is greater than the sample mean. We propose in this paper a hierarchical log-normal Poisson model with a Lasso penalty, which has the advantage of directly modelling inter-sample variability and can therefore be readily applied to the raw data. Performance of these different methods for gene network inference are compared on data simulated under a multivariate Poisson distribution (Karlis and Meligkotsidou, 2005) with various amounts of additional inter-sample variability, as well as on publicly available microRNA-seq data collected on breast invasive carcinoma (BRCA) tumors, downloaded from The Cancer Genome Atlas (TCGA) Data Portal.

7.2 Materials and Methods

We first define the notation that will be used throughout this paper. Let Y_{ij} be the random variable corresponding to the gene expression measure for the sample i $(i = 1, \ldots, n)$ for the gene j $(j = 1, \ldots, p)$, with y_{ij} being the corresponding observed value of Y_{ij}. Note that i always indexes samples and j always indexes genes with n the number of samples and p the number of genes. A network represents gene interactions. The nodes are random variables modelling the gene expression levels and the edges indicate the dependencies between those variables. In this section we provide a short description of the models that will be compared for gene network inference from RNA-seq data.

7.2.1 Gaussian graphical model

The underlying assumption of this model is that the data are normally distributed. In the case of untransformed RNA-seq data, this assumption is not valid since data counts cannot take negative values. We investigated a variety of Box-Cox transformations to lead to approximately normal data (Box and Cox, 1964), where the δ value was chosen to maximize the log-likelihood of the transformed data :

$$y_{ij} \to f(y_{ij}) = \begin{cases} \dfrac{y_{ij}^{\delta} - 1}{\delta}, & \text{if } \delta \neq 0, \\ \log(y_{ij}), & \text{if } \delta = 0. \end{cases}$$

Since gene expression data may contain zero counts, we usually use $(y+1)$ instead of y in the Box-Cox formula above. Let $\mathbf{z}_i = (f(y_{i1}), ..., f(y_{ip}))$ be the transformed vector of expression values for p genes for the ith biological sample $(i = 1, ..., n)$. We assume that $\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The edges of the inferred network correspond to non-zero partial correlations, i.e. the non-zero elements of matrix $\boldsymbol{\Sigma}^{-1}$ (Whittaker, 1990; Friedman et al., 2008).

Let $\boldsymbol{S}$ be the empirical covariance matrix. The log-likelihood of the model is :

$$L(\boldsymbol{\Sigma^{-1}}) = \log(\det(\boldsymbol{\Sigma^{-1}})) - \text{trace}(\boldsymbol{S\Sigma^{-1}}). \tag{7.1}$$

A common assumption in the context of gene networks is that the matrix Σ^{-1} is sparse. We add an ℓ_1 penalty to the log-likelihood (8.1) so that some coefficients in the estimated Σ^{-1} matrix are precisely equal to 0 :

$$\log(\det(\boldsymbol{\Sigma^{-1}})) - \text{trace}(\boldsymbol{S\Sigma^{-1}}) - \lambda\|\boldsymbol{\Sigma^{-1}}\|_{\ell_1}. \tag{7.2}$$

Network inference using a Gaussian graphical model has been extensively studied and used over the past years. Many methods exist to compute the penalized maximum likelihood estimate of the $\boldsymbol{\Sigma}$ matrix above. We use the method implemented in the `glasso` R package (Friedman et al., 2008) which makes use of a coordinate descent algorithm.

The choice of the regularization parameter λ has also been extensively studied (Giraud et al., 2012). We choose to perform model selection by maximizing the Bayesian

Information Criterion (BIC) (Schwarz, 1978) defined below, where ν represents the number of free parameters in the model :

$$\text{BIC} = L(\mathbf{\Sigma^{-1}}) - \nu\frac{\log n}{2}. \tag{7.3}$$

Note that a single parameter λ is chosen for the entire network.

7.2.2 Log-linear Poisson graphical model

A log-linear Poisson graphical model specifically designed for network inference from count data has been recently proposed (Allen and Liu, 2012). This model is based on a Poisson distribution which assumes the mean and variance to be equal. Therefore, the model does not account for the high dispersion of the data, also called over-dispersion with respect to the Poisson distribution, when the sample variance is higher than the sample mean. To apply it to RNA-seq data, the authors propose to use a power transformation of the data $y_{ij} \to g(y_{ij}) = y_{ij}^{\alpha}$, with $\alpha \in]0, 1]$ implemented in the R package `PoiClaClu` (Li et al., 2012). The coefficient α is chosen to maximize an adequacy criterion between the transformed data $\mathbf{y}^{\alpha}$ and a Poisson distribution.

Let $\boldsymbol{z}_j = (g(y_{1j}), ..., g(y_{nj}))$ be the transformed vector of expression values for gene j in the n biological samples. It is assumed that the conditional distribution of Z_{ij} given all the other genes $\mathbf{z}_{i(-j)} = (z_{i,1}, \ldots, z_{i(j-1)}, z_{i(j+1)}, \ldots, z_{ip})$ is a Poisson distribution $\mathcal{P}(\mu_j)$, with $\log(\mu_j)$ modelled as a linear regression on all the other genes :

$$p(Z_{ij}|\mathbf{z}_{i(-j)}) \sim \mathcal{P}(\mu_j)$$

with

$$\log(\mu_j) = \sum_{j' \neq j} \beta_{jj'} \tilde{z}_{ij'}.$$

The notation $\tilde{\boldsymbol{z}}$ corresponds to a standardization of the log-transformed data. This standardization is a necessity since we model the mean of the gene j and not the random variable itself. An edge is present in the inferred graph if one or both parameters $\beta_{jj'}$ and $\beta_{j'j}$ are different from zero. The log-likelihood for gene j can be written in this case as :

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[z_{ij} \exp(\sum_{j \neq j'} \beta_{jj'} \tilde{z}_{ij'}) - \sum_{j \neq j'} \beta_{jj'} \tilde{z}_{ij'} \right]. \tag{7.4}$$

Similar to the previous model, we assume that the vector $\boldsymbol{\beta}_j$ is sparse. We add an ℓ_1 penalty to the log-likelihood (7.4) so that some coefficients in the estimated $\boldsymbol{\beta}_j$ vector are set to 0. Estimation of parameters $\boldsymbol{\beta}_j$ can be obtained by a coordinate gradient algorithm as implemented in the R package `glmnet` (Friedman et al., 2010). We propose to perform the model selection with the Stability Approach to Regularization Selection criterion (StARS), as suggested by Allen and Liu (2012). This stability-based method selects the network with the smallest amount of regularization that simultaneously makes the network sparse and replicable under random sampling. Note that we select only one regularization parameter for all the regressions in the network problem.

7.2.3 Hierarchical log-normal Poisson graphical model

We note that the Poisson model presented above requires a transformation of the data to account for the high dispersion. Here we propose to deal with it directly with a hierarchical log-normal Poisson model. The count expression of gene j for sample $i \in 1, \ldots, n$ is modeled as : $Y_{ij} \sim \mathcal{P}(\theta_{ij})$ with

$$\log(\theta_{ij}) = \sum_{j' \neq j} \beta_{jj'} \tilde{y}_{ij'} + \varepsilon_{ij},$$

$$\boldsymbol{\varepsilon_j} = (\varepsilon_{1j}, ..., \varepsilon_{nj}) \sim \mathcal{N}(0, \sigma_j^2 \boldsymbol{I_n}).$$

As before, the notation $\tilde{\boldsymbol{y}}$ corresponds to a standardization of the log-transformed data. Here, the vector $\mathbf{Y}_j \sim \mathcal{P}(\boldsymbol{\theta_j})$ and $\boldsymbol{\theta_j}$ is itself a random variable : $\boldsymbol{\theta_j} = \mu_j \exp(\boldsymbol{\varepsilon_j})$ with $\boldsymbol{\varepsilon_j} \sim \boldsymbol{N}_n(0, \sigma_j^2 I_n)$ and $\mu_j = \exp(\sum_{j' \neq j} \beta_{jj'} \tilde{y}_{ij'})$. Note that the variance of the random variable $\mathcal{P}(\boldsymbol{\theta_j})$ is larger than its mean if σ_j^2 is positive. As previously, an edge is present in the graph between genes j and j' if one or both parameters $\beta_{jj'}$ and $\beta_{j'j}$ are different from zero.

In this model, the likelihood for gene j can be written as :

$$L(\boldsymbol{\beta_j}, \sigma_j) = \int_{\mathbb{R}} \left(\prod_{i=1}^{n} \Big[\exp(-\mu_{ij} + y_{ij} \log(\mu_{ij}) - \log(y_{ij}!) \Big] \frac{1}{(2\pi)^{n/2} \sigma_j^n} \exp\left(-\frac{1}{2\sigma_j^2} ||\boldsymbol{\varepsilon_j}||_2^2 \right) \right) d\boldsymbol{\varepsilon_j}. \tag{7.5}$$

Similar to the previous model, we assume that the vector $\boldsymbol{\beta}_j$ is sparse. We add an ℓ_1 penalty to a function of the log-likelihood (7.5) so that some coefficients in the estimated $\boldsymbol{\beta}_j$ vector are set to 0 :

$$-2L(\boldsymbol{\beta_j}, \sigma_j) + \lambda \|\boldsymbol{\beta_j}\|_{\ell_1}.$$

Estimation of parameters $\boldsymbol{\beta_j}$ and σ_j was done using the R function `glmmixedlasso` (Schelldorfer et al., 2014), based on a Laplace approximation of the penalized likelihood and a coordinate descent algorithm.

An important aspect of this method is the choice of the regularization parameter λ. To choose a common λ parameter for all the gene-by-gene regressions, we propose to use a two stage approach for this parameter. First, for each gene j, a λ_j parameter is chosen by maximizing the BIC criterion defined as $\text{BIC} = L(\boldsymbol{\beta_j}, \sigma_j) - \nu \log(n)/2$, where $L(\boldsymbol{\beta_j}, \sigma_j)$ is the unpenalized log-likelihood and ν is the number of free parameters in the model. Then the mean of the λ_j parameters is taken as the regularization parameter and used for all the regressions : $\lambda = \sum_{j=1}^{p} \lambda_j / p$. Since BIC is an asymptotic criterion, taking the average of the regularization parameters over all the regressions helps to improve network inference performance.

7.3 Results

7.3.1 Simulation study

Multivariate Poisson data simulation

In order to simulate multivariate Poisson data, we use a method described by Karlis (Karlis and Meligkotsidou, 2005). As an illustration, for a two dimensional multivariate Poisson distribution, we simulate three independent Poisson variables (X_1, X_2, X_{12}) and sum them up ($Y_1 = X_1 + X_{12}$ and $Y_2 = X_2 + X_{12}$) so that the resulting variables are not independent : $\text{cov}(Y_1, Y_2) \neq 0$ if $E(X_{12}) \neq 0$. In the general case, a sample $\mathbf{y}$ of dimension $(n \times p)$ where p is the number of nodes in the network, n the number of samples is obtained by summing samples from $(p + p(p-1)/2)$ independent Poisson random variables. The adjacency matrix $\boldsymbol{A} \in \{0,1\}^{p \times p}$ encodes the underlying graph structure : $A_{ij} = 0$ means that the expression level of genes $i \in 1, \ldots, p$ and $j \in 1, \ldots, p$ are conditionally independent given the other gene expression levels. In order to sum the $(p + p(p-1)/2)$ terms accordingly, we fix the matrix $\boldsymbol{B}$ of dimension $(p \times (p + p(p-1)/2)$: $\boldsymbol{B} = [\boldsymbol{I}_p; \boldsymbol{P} \odot (\boldsymbol{I}_p \text{tri}(\boldsymbol{A})')]'$ where $\boldsymbol{P}$ is a permutation matrix of dimension $(p \times (p(p-1)/2))$ of vector (1,1,0,...,0), $\odot$ denotes the matrix multiplication element by element and $\text{tri}(\boldsymbol{A})$ is the vector of dimension $(p(p-1)/2) \times 1$ containing the elements of the upper triangular adjacency matrix. The matrix product $\mathbf{y} = \boldsymbol{BX}$ gives a count data table of size $n \times p$: n samples from a p-dimensional Poisson random variable whose underlying dependency structure is encoded in the known $\boldsymbol{A}$ matrix.

RNA-seq data are known to be overdispersed relative to a Poisson distribution with the sample variance of a gene expression vector larger than the sample mean. In our simulation study, we also consider the possibility of inflating the variance of the independent Poisson random variables used in the $\boldsymbol{X}$ matrix of the formula above by simulating independent variables according to a log-normal Poisson model. For gene j and sample i, we sample $X_{ij} \sim \mathcal{P}(\mu_{ij})$ with $\log(\mu_{ij}) = \theta_j + \varepsilon_{ij}$, $\varepsilon_{ij} \sim \mathcal{N}(0, \sigma_j^2)$. We use this log-normal Poisson distribution only for the first p columns of the matrix, the other columns being sampled from a simple Poisson distribution.

Simulation settings

The three methods were compared on two sets of simulations : multivariate Poisson data and overdispersed multivariate Poisson. For each type of data, we simulated 50 different adjacency matrices $\boldsymbol{A}$ with a scale-free structure. This implies that degrees of the edges are assumed to follow a power law distribution, i.e. few nodes in the network are well connected and most of the nodes have only one or two neighbours. The number of nodes p was set to 50. With a scale-free structure, the maximum degree of a node is $k_{max} = 35$ and the average degree is less than 2. To avoid the ultra-high dimensional setting, defined as $k \log(\frac{p}{k})/n \geq \frac{1}{2}$ for Gaussian linear regression (Verzelen, 2012), we set the number of biological samples to $n = 100$. For each of the 50 different adjacency matrices, 1225 samples of size n were simulated from Poisson random variables (adding extra inter-sample variance or not) and summed up as explained above to obtain the final data set of size 100×50. We chose to use Poisson distributions of mean $\mu = 100$ to

build the $\boldsymbol{X}$ data matrix, resulting in data counts ranging from around 100 to 2500. In the case of Poisson data with inflated variance, the parameter σ_j was set to 0.25, which is slightly smaller than the amount of dispersion observed in the real data presented below.

To evaluate the different methods, we tried to infer the adjacency matrix $\boldsymbol{A}$ from the simulated dataset $\mathbf{y}_{(100\times50)}$ and compared the inferred matrix $\boldsymbol{A}_{pred}$ with the real adjacency matrix $\boldsymbol{A}$ used to simulate the data. For each type of data (with and without extra inter-sample variance) and for each network inference method (Gaussian, log-linear Poisson, and the proposed hierarchical log-normal Poisson graphical models), Receiver Operating Characteristic (ROC) curves were constructed by varying values of the regularization parameter from an empty network (sensitivity equal to 0) to a full network (specificity equal to 0). The sensitivity and specificity values were also compared for the different methods using the chosen regularization parameter (with the BIC criterion for the Gaussian graphical model, StARS criterion for the log-linear Poisson graphical model and the mean-BIC criterion presented above for the hierarchical log-normal Poisson model). Note that in the case of the Poisson graphical model, a power transformation is applied only in the simulation setting inducing inflated variance.

Results

ROC curves, averaged over the 50 simulated datasets, are presented in Figure 7.1 for the two simulation settings (multivariate Poisson data with or without inflated variance). It can be noticed that in the first setting, with no over-dispersion, the log-linear Poisson model outperforms the Gaussian graphical model applied to transformed data. This result was already observed (Allen and Liu, 2012). As expected, in this case the performance of the log-linear Poisson model and the proposed hierarchical model are very similar. When adding extra variability to the data, we are compelled to use a power-transformation of the data to apply the log-linear Poisson model (Allen and Liu, 2012), since the data no longer respect the Poisson assumption of equal mean and variance. The performance of the log-linear Poisson model in this case is considerably deteriorated, and is now comparable to the poor performance of the Gaussian graphical model on log-transformed data. The proposed hierarchical log-normal Poisson model therefore outperforms the two other methods in this case, keeping in mind that the data were simulated under a closely related model that was deemed to be a reasonable choice to approximate the dynamics of RNA-seq data. It has to be pointed out that for the over-dispersed data, performances of the three methods are considerably worse compared to the simple case of multivariate Poisson data due to the presence of additional variability.

Sensitivity and specificity obtained by each method for the chosen regularization parameters are represented in diamond-shape squares on the ROC curves (Figure 7.1) and are summarized in Table 7.1.

The regularization parameter chosen with the mean-BIC criterion for the proposed hierarchical log-normal Poisson model offers a higher sensitivity than the Poisson or Gaussian graphical models, even when no over-dispersion was simulated (0.84 compared to 0.71 and 0.57, respectively), while keeping a high specificity (0.97 compared to 0.99 and 0.98, respectively). The number of correctly detected edges is therefore larger for the proposed model compared to the other two methods, even in the case of multivariate

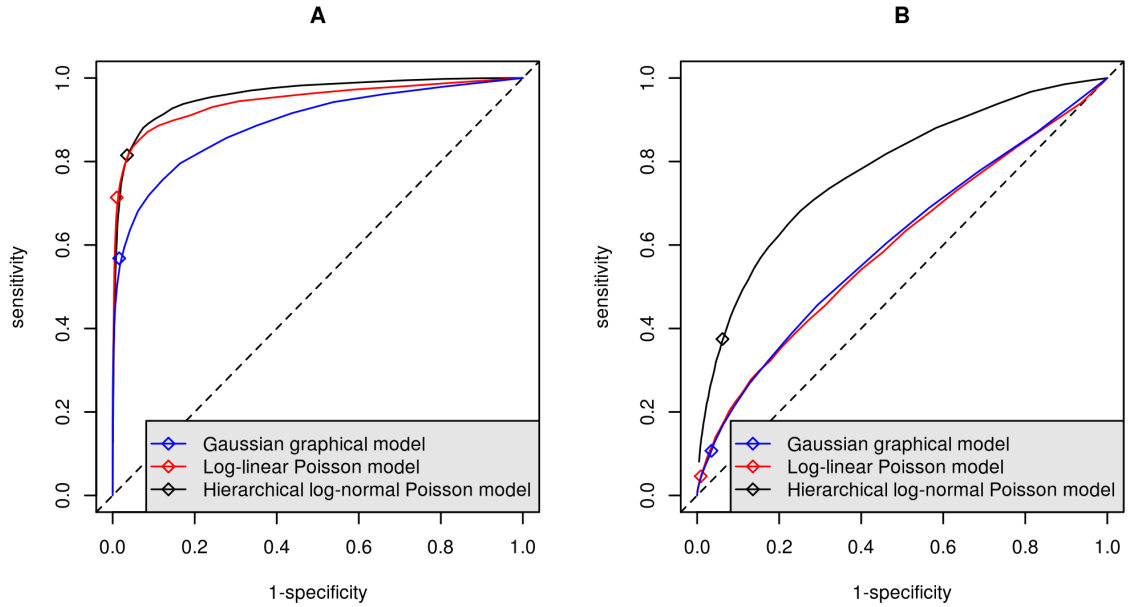

FIGURE 7.1 – **ROC curves, averaged over 50 simulated data sets on scale-free graphs.** Results are presented for the Gaussian graphical model on log-transformed data (blue), the log-linear Poisson graphical model on power-transformed data (red) and the hierarchical log-normal Poisson model on raw data (black) on multivariate Poisson data (A) and multivariate Poisson data with inflated variance (B). The dotted black lines represent the diagonals.

		GGM	Log-linear Poisson	Hierarchical model
Multivariate	Sens.	0.568 (0.069)	0.714 (0.036)	0.838 (0.050)
Poisson Data	Spec.	0.984 (0.003)	0.990 (0.003)	0.967 (0.006)
Over-dispersed	Sens.	0.107 (0.045)	0.046 (0.033)	0.383 (0.064)
Poisson Data	Spec.	0.965 (0.003)	0.991 (0.004)	0.982 (0.027)

TABLE 7.1 – **Average sensitivity and specificity (standard deviation in parentheses) for the selected network across 50 simulated networks with scale-free structure.** Results are averaged over 50 datasets for multivariate Poisson data and over-dispersed multivariate Poisson data. GGM : Gaussian graphical model on transformed data (log(y+1)), Log-linear Poisson : log-linear Poisson graphical model proposed by (Allen and Liu, 2012) on power transformed data (y^{α}), Hierarchical model : proposed model as detailed in the Methods section and applied on the raw data.

Poisson data with no over-dispersion. When adding extra inter-sample variability, the differences between the three methods are even larger, even if the performances deteriorate for all methods (sensitivity equal to 0.4 for the proposed model compared to 0.1 for the Gaussian graphical model and 0.05 for the Poisson graphical model). These very low sensitivity values can partly be explained by the fact that scale-free structures were considered for the simulated graphs, therefore generating only a small number of edges compared to a random graph structure that are difficult to correctly detect. This also explains, on the other hand, the high specificity values. In fact, as the models infer very few edges for low numbers of biological replicates, they have less chance to detect incorrect edges. Both the ROC curves and the sensitivity/specificity for the chosen regularization

		GGM	Log-linear Poisson	Hierarchical model
Multivariate	Sens.	0.571 (0.059)	0.691 (0.061)	0.763 (0.093)
Poisson Data	Spec.	0.992 (0.003)	0.990 (0.003)	0.975 (0.005)
Over-dispersed	Sens.	0.112 (0.065)	0.050 (0.041)	0.198 (0.060)
Poisson Data	Spec.	0.971 (0.003)	0.990 (0.003)	0.958 (0.009)

TABLE 7.2 – **Average sensitivity and specificity (standard deviation in parentheses) for the selected network across 30 simulated networks with Erdos-Rényi structure.** Results are averaged over 30 datasets for multivariate Poisson data and overdispersed multivariate Poisson data. GGM : Gaussian graphical model on transformed data (log(y+1)), Log-linear Poisson : log-linear Poisson graphical model proposed by (Allen and Liu, 2012) on power transformed data (y^{α}), Hierarchical model : proposed model as detailed in the Methods section and applied on the raw data.

parameter therefore show much better performances for the proposed hierarchical model than the Gaussian graphical model on log-transformed data or the Poisson graphical model on power-transformed data, especially in the case of overdispersed multivariate Poisson data.

Figure 7.2 represents the relationships between the degree of the nodes in the estimated network and in the simulated structure for both the Poisson graphical model and the proposed hierarchical model. It can be observed that, as expected, in the case of no over-dispersion, both methods perform quite similarly, as already seen in the ROC curves above. In the case of over-dispersion, however, even if the sensitivity was quite poor for all methods (Table 7.1), the structure of the graph was much better preserved with the proposed model than with the Poisson graphical model on power transformed data.

To ensure that these results do not depend on the scale-free structure of the graphs, we have drawn ROC curves and performed similar model selection on data simulated with an Erdos-Rényi structure (Erdos and Rényi., 1959) (Figure 7.3 and Table 7.2). For Erdos-Rényi graphs, each pair of nodes are connected with the same probability, independently of the other pairs of nodes. As previously observed (Allen and Liu, 2012), although the differences among the three methods are less pronounced for Erdos-Rényi structures than for scale-free structures, the same general conclusions hold.

7.3.2 Real data analysis

Data description

The three methods were applied to a publicly available microRNA-seq data set available at The Cancer Genome Atlas (TCGA) Data Portal (`http://cancergenome.nih.gov/`). We selected 100 samples from breast invasive carcinoma (BRCA) tumors. To avoid being in an ultra high-dimensionality setting (Verzelen, 2012), we reduced the number of microRNAs used for network inference to 50 (among 863). To do so, we first removed all microRNAs that had at least one null count. Among the remaining 207, we selected the microRNAs with the largest inter-sample variance (as suggested by (Allen

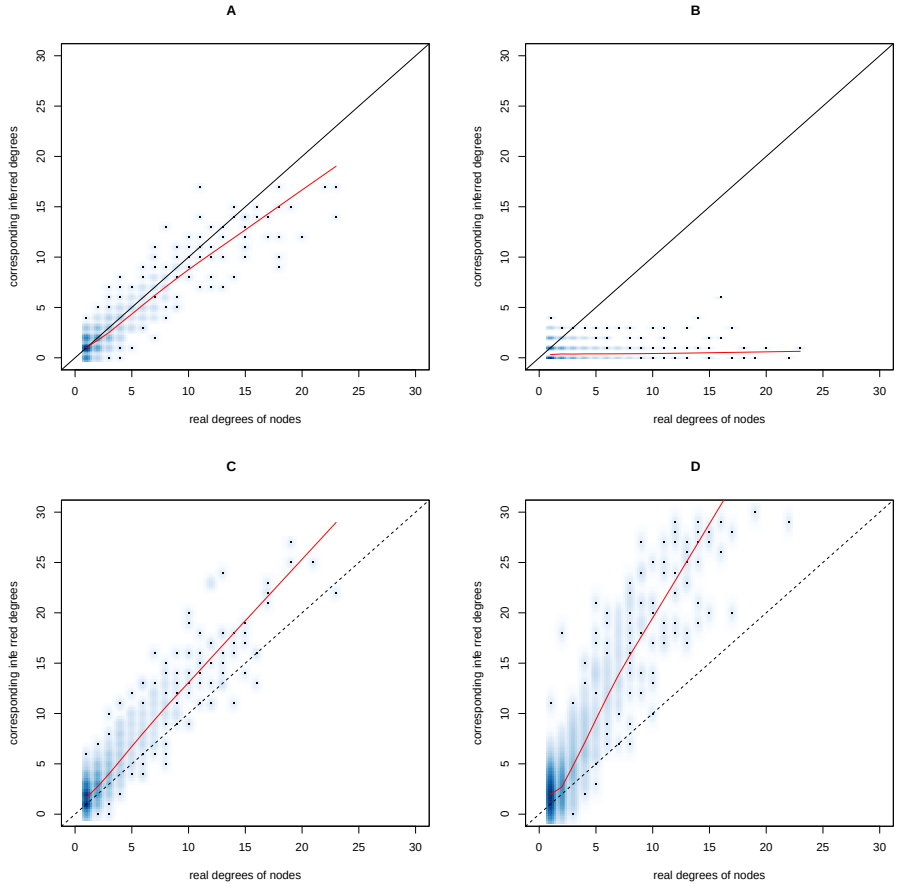

FIGURE 7.2 – **Relationship between the degree of the nodes in the estimated network and in the simulated network.** Results are presented for the log-linear Poisson graphical model without over-dispersion (A) and with over-dispersion (B), for the proposed hierarchical log-normal Poisson graphical model without over-dispersion (C) and with over-dispersion (D). Black dotted lines represent the diagonal, and red lines represent loess curves.

and Liu, 2012)). These microRNAs are the most likely to be linked to breast cancer development since they are selected among the most highly variable microRNAs. Note that we did not perfom any normalization for differences in library sizes on this data set, as contrary to differential analyses (Anders and Huber, 2010; Robinson and Oshlack, 2010), differences in library sizes have no impact on the network inference results since we do not compare two different biological samples, but relate the expression of genes within each biological sample. Since each miRNA has an equal number of nucleotides, there is no need for a gene length correction either.

Modelling the data

Shapiro-Wilk tests on miRNA expression vectors showed that the data, even for highly expressed miRNAs, could not be directly modelled as a normal distribution (Shapiro S. S. and Wilk M. B., 1965). We therefore used a Box-Cox transformation (Box and Cox, 1964) prior to applying a Gaussian graphical model to these data. The optimal

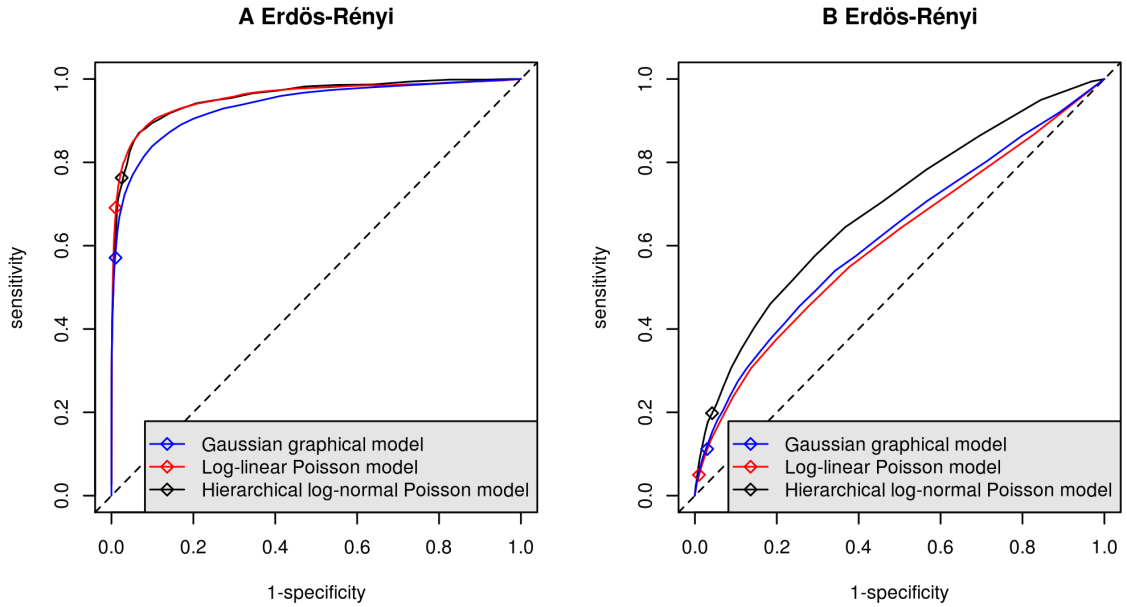

FIGURE 7.3 – **ROC curves, averaged over 30 simulated data sets on Erdos-Rényi graphs.** Results are presented for the Gaussian graphical model on log-transformed data (blue), the log-linear Poisson graphical model on power-transformed data (red) and the hierarchical log-normal Poisson model on raw data (black) on multivariate Poisson data (A) and multivariate Poisson data with inflated variance (B). The dotted black lines represent the diagonals.

Box-Cox parameter to make the data as normally distributed as possible was found to be close to zero, which corresponds to a log-transformation of the data (Figure 7.4).

For these data, the Poisson assumption is not verified either, as shown in Figure 7.5, since the sample variance is considerably larger than the sample mean for all miRNAs. As suggested in (Allen and Liu, 2012), we therefore applied the power-transformation implemented in the `PoiClaClu` package prior to applying the log-linear Poisson graphical model.

The Gaussian graphical model with the BIC criterion detected 48 edges, the log-linear Poisson graphical model with the StARS criterion (Allen and Liu, 2012) detected 74 edges, and the proposed hierarchical log-normal Poisson graphical model detected 369 edges among the 50 miRNAs considered here. As shown in Figure 7.5, these data exhibit significant over-dispersion with respect to the Poisson assumption. We are therefore close to the second simulation setting presented above. In this case, the sensitivity of the proposed hierarchical model is expected to be much higher than for the other two methods, which explains the much larger number of detected edges. Figure 7.6 presents the network inferred by the hierarchical model. Table 7.3 presents the biological functions of the most highly connected nodes found with the proposed hierarchical model. It can be noticed that a large majority of these miRNAs are already known to be related to breast cancer. Further biological validation would be interesting for the remaining ones that could be new potential therapeutic targets.

7.4 Discussion

Network inference from RNA-seq data is an important methodological challenge. This work is a pioneer study to provide some guidelines on the best methods to achieve this goal. There are two main approaches. The first and simplest idea is to perform a transformation of the data and apply previously proposed methods for microarray studies based on Gaussian graphical models, for example using a Box-Cox transformation. Another possibility is to apply methods specifically developed for the analysis of count data using Poisson graphical models, either with a power transformation of the data or by accounting for over-dispersion directly in the model using for example a hierarchical log-normal Poisson graphical model as proposed here. We found in both simulation study and real data application that the power transformation did not work well to correct for over-dispersion. It has to be noted that the same α parameter was used here for all the genes. It might be possible to improve the performance of this method if a different coefficient was estimated for each gene. This is, however, not possible with the method proposed by (Witten, 2011), which finds the optimal value by maximizing the adequacy criterion for a group of genes. In this work the best suited methodology for network inference from RNA-seq data currently appears to be the proposed hierarchical Poisson log-normal model, which seems to be able to appropriately deal with highly dispersed count data. However, the implementation of this approach based on the R package `glmmixedlasso` (Schelldorfer et al., 2014) is quite slow for a large number of biological samples and more research is needed to optimize this function.

It has to be pointed out that in high-dimensional settings (number of genes much larger than the number of biological samples), all methods were unsurprisingly found to perform very poorly, despite the ℓ_1 regularization. As for microarray studies, the limited number of biological replicates available in RNA-seq experiments considerably restrains the number of genes that can be included in the network. Future research is needed to tackle this issue. A first possibility may be to try to reduce the number of parameters to be estimated. In fact, in a first step we aim at finding the regulatory relationships between genes without necessarily estimating their strength precisely. Therefore, in the regression models presented above, instead of trying to estimate one parameter for each gene we could infer parameters for groups of genes. Alternatively, to face the problem of small numbers of biological replicates, instead of inferring regulatory networks within each experimental condition, it would be interesting to use joint graphical model approaches (Guo et al., 2011) to jointly infer a network in multiple conditions, thus highlighting the common or differing patterns across conditions.

miRNA	reference
hsa-mir-451	BC (Kovalchuk et al., 2008)
hsa-let-7b	BC (Peter, 2009)
hsa-mir-486	BC (Dalmay and Edwards, 2006)
hsa-let-7f-2	cancer (Peter, 2009)
hsa-mir-150	no reference
hsa-mir-145	BC (Zou and Xu, 2012)
hsa-mir-24-2	BC (Srivastava et al., 2011)
hsa-mir-200c	BC (Gregory et al., 2008), (Peter, 2009)
hsa-mir-143	BC (Stahlhut Espinosa and Slack, 2006)
hsa-mir-142	no reference

TABLE 7.3 – **Ten most highly connected genes in the network inferred by the proposed hierarchical model. BC corresponds to miRNAs known to be linked to Breast Cancer, with the corresponding references.**

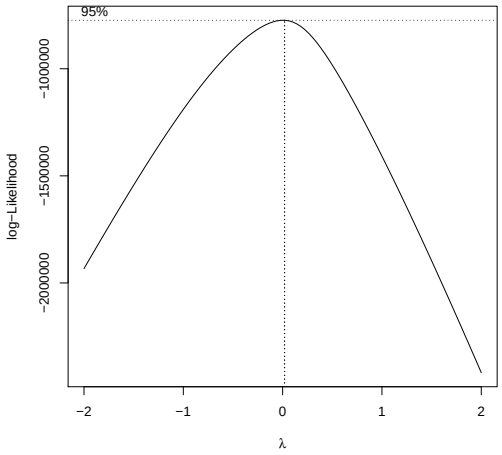

FIGURE 7.4 – **Optimal parameter for the box-cox transformation of data.** Curve obtained with the R package `MASS`.

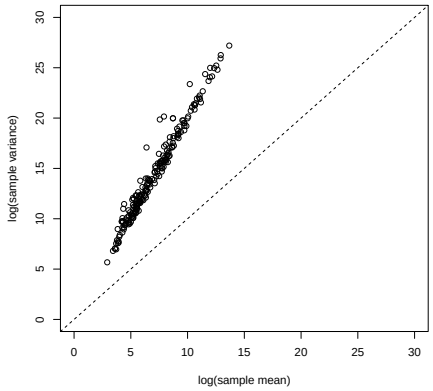

FIGURE 7.5 – **Sample mean-variance relationship for the 207 microRNAs.**

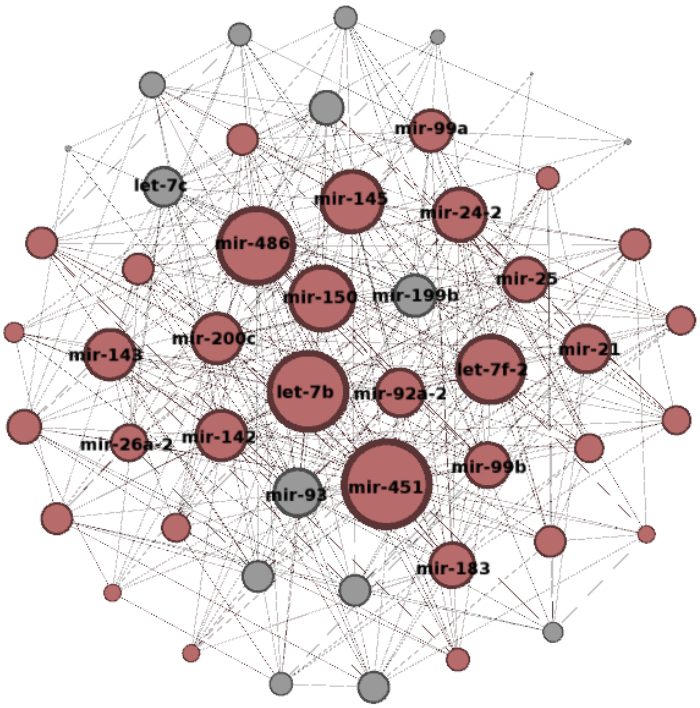

FIGURE 7.6 – **Network inferred with the hierarchical model.** The representation was obtained using the software `Gephi` (Bastian et al., 2009). The size of nodes represents the number of edges associated with the corresponding gene in the network.

Chapitre 8

Block diagonal covariance selection for gaussian graphical model in high dimension

Résumé. Les modèles graphiques gaussiens permettent d'inférer et de visualiser les dépendances entre des variables. Ces modèles sont difficiles à estimer lorsque la taille de l'échantillon est plus petite que le nombre de variables. Afin de réduire la dimension du problème, nous proposons une procédure de sélection de modèle non-asymptotique : nous approchons la matrice de covariance par une matrice diagonale par blocs. Pour détecter la structure de cette matrice, nous seuillons la matrice de covariance empirique, le seuil étant choisi à l'aide de l'heuristique de pente. Grâce à cette structure, le problème d'estimation est décomposé en plusieurs sous-problèmes indépendants : dans chaque bloc, nous inférons les dépendances entre variables à l'aide du graphical Lasso. Nous justifions cette procédure par des résultats théoriques. Nous illustrons aussi cette méthode sur des données simulées et des données réelles.

Abstract. Gaussian graphical models provide networks to recover and visualize dependencies between continuous variables. Inferring the graph is difficult when the sample size is smaller than the number of variables. To reduce the number of parameters to estimate in the model, we propose a non-asymptotic model selection procedure : we approximate the covariance matrix of the model by a block diagonal matrix. To detect the structure of this matrix, we threshold the sample covariance matrix, choosing the threshold using the slope heuristic. Subsequently, the estimation problem is divided into several independent problems based on the block diagonal structure : in each block, we estimate the network of dependancies using the Graphical lasso algorithm. We justify our procedure from a theoretical point of view and illustrate it on simulated and real datasets.

8.1 Introduction

We are interested in designing a Gaussian graphical model (GGM) from n observations described with p variables in a situation where p is large and n small in comparison to p. Let $\mathbf{y}$ be the centered and scaled data matrix of size $n \times p$, y_{ij} denoting the value of observation i for variable j. The vectors $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$ are assumed to be independent and to follow a $p-$multivariate normal distribution $\phi(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Sigma}_{j,j} = 1$ for each $j \in \{1, \ldots, p\}$. The edges of the corresponding graph are the non-zero coefficients of the matrix $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$, estimated by the graphical lasso defined by (8.1), where S is the empirical covariance matrix. Null coefficients in the inverse covariance matrix $\mathbf{\Theta}$ indicate conditionally independent variables (Whittaker, 1990). To estimate $\mathbf{\Theta}$ and recover the graph, we seek to maximise the following penalized log-likelihood :

$$\begin{aligned} \ell_\lambda(\mathbf{\Theta}) &= \log\det(\mathbf{\Theta}) - \text{trace}(S\mathbf{\Theta}) - \lambda \mid\mid \mathbf{\Theta} \mid\mid_1 \\ \hat{\mathbf{\Theta}}^{(\lambda)} &= \underset{\mathbf{\Theta} \in \mathbb{S}_p^{++}}{\text{argmin}} \left\{ \ell_\lambda(\mathbf{\Theta}) \right\}, \text{ for } \lambda \geq 0, \end{aligned} \tag{8.1}$$

where $\mathbb{S}_p^{++}(\mathbb{R})$ is the set of symmetric and positive matrices on $\mathbb{R}$.

A standard algorithm to solve this problem is the graphical lasso implemented in the R `glasso` package (Friedman et al., 2008). Even if the graphical lasso is designed to solve high-dimensional problems ($n < p$), the graphical lasso performs poorly when the number of observations n is too small (Giraud, 2008; Verzelen, 2012). Besides, Mazumder and Hastie (2012) and Witten et al. (2011) have shown the following property of the graphical lasso solution : if the solution $\hat{\mathbf{\Theta}}^{(\lambda)}$ of the problem (8.1) has a block diagonal structure for some ordering of the variables, the inference problem can be solved in each block independently. We always consider block diagonal structure for some ordering of the variables. Based on the sub gradient equations used in the graphical lasso algorithm (Friedman et al., 2008), there is an easy way to check if the solution $\hat{\mathbf{\Theta}}^{(\lambda)}$ will have a block diagonal structure : the solution $\hat{\mathbf{\Theta}}^{(\lambda)}$ and the thresholded sample covariance matrix $S^{(\lambda)} = (\hat{\mathbf{\Sigma}}_{jj'} \mathbf{1}_{\{|\hat{\mathbf{\Sigma}}_{jj'}| > \lambda\}})_{jj'}$ have the same block diagonal structure. This result induces the following *block diagonal screening rule* to speed up computations and improve estimation in the graphical lasso. For a given λ, solving the optimization problem in (8.1) amounts to :

Step 1 : identify of the connected components of the graph encoded in the adjacency matrix $\left(\mathbf{1}_{\{|\hat{\mathbf{\Sigma}}_{jj'}| > \lambda\}}\right)_{jj'}$,

Step 2 : solve the graphical lasso problem with regularization parameter λ in each connected component, independently.

This *block diagonal screening rule* has been included by default in the last version of the `glasso` package (Friedman et al., 2008). It substantially reduces the algorithm complexity from $O(p^3)$ to $O(p^2 + \sum_{k=1}^{K} p_k^3)$ where p_k is the number of variables in block k. This reduction is of great interest and has been used for sparse quadratic discriminant analysis (Le and Hastie, 2014). We note that the parameter λ used for thresholding the matrix $\mid S \mid$ in step 1 is the same parameter λ used for regularization in each graphical lasso sub-problem in step 2.

Tan et al. (2015) proposed a variation of this two-step procedure, called the *Cluster Graphical Model* : they detect the connected components by clustering the variables into K groups using a complete linkage clustering on the sample covariance matrix $\tilde{S} = \mid S \mid$,

and then, infer a network on each group independently, each with its own regularization parameter λ. Their numerical results suggest that choosing different parameters λ for step 1 and for each subproblem in step 2 improves network inference performance. They choose the number of blocks K (which is equivalent to the selection of threshold λ for $S^{(\lambda)}$) using a leave-one-out algorithm to recast the unsupervised clustering into a supervised one and select the number K giving the smallest mean square error. They also argue that the method is not sensitive to the number of clusters.

In the spirit of the Cluster Graphical Model, we propose to solve the network inference problem using different parameters λ for step 1 and sub-problems in each block for step 2. We pay particular attention to the detection of the connected components in step 1 of the procedure and turn this detection into a model selection problem. We construct a collection of models with different block diagonal structures for the covariance matrix $\mathbf{\Sigma}$, obtained by thresholding the sample covariance matrix S. Then, we choose a model among this collection using a model selection criterion.

Akaike (1974) and Schwarz (1978) have introduced penalized model selection criteria. These criteria are based on asymptotic approximations which are not ensured to hold for small values of n. More recently, Birgé and Massart (2007) have introduced a non-asymptotic criterion : the slope heuristic. The slope heuristic is especially useful when n is small with respect to the model dimension and when the model family is somewhat biased with respect to the sampling distribution. Baudry et al. (2012) have provided some practical tools to implement the slope heuristic from Birgé and Massart (2007). It has proven to be effective in a variety of practical situations : for exemple, Rau et al. (2015) select the number of components in Poisson mixture models on gene expression data using the slope heuristic. Bouveyron et al. (2015) select the number of components in discriminative functional mixture model on data describing bike sharing systems using the slope heuristic. However, the theoretical justification of the slope heuristic has met with several technical difficulties. Birgé and Massart (2007) and Baraud et al. (2009) prove the existence of minimal penalties in heteroscedastic regression with fixed design, Arlot and Massart (2009) extends it for homoscedastic regression with fixed design. Few papers provide theoretical guarantees : Lebarbier (2005) for multiple change point detection, Castellan (2010) for selecting the best partition for histogram construction and Maugis and Michel (2011) for variable selection in mixture models.

In the context of graphical models with block diagonal structure, we provide an oracle inequality to provide an upper bound for the risk between the true model and the model selected among the collection of models with different block diagonal structures. We also provide a lower bound of this same risk. This oracle inequality theoretically supports our model selection procedure based on the slope heuristic with a penalty proportional to the dimension (up to a logarithm term). The main difficulty to prove these theoretical results arises from the fact that the family of models to be considered is random. More precisely, the collection of the block diagonal structures obtained by thresholding is data-driven, and therefore random. Our oracle inequality is derived from a theorem developed in Massart (2007). For the lower bound of the risk, some results have been previously obtained by Bickel and Levina (2008) or (Cai et al., 2010). To obtain our lower bound, we use the Birgé lemma developed in Birgé (2005) in conjunction with a discretization of the model collection space.

The chapter is organized as follows. In Section 8.2, after providing basic notations

and definitions, the procedure we propose to detect the block-diagonal structure by non-asymptotic model selection is described. Section 8.3 describes theorical results supporting our model selection procedure. Section 8.4 investigates the numerical performance of our procedure in a simulation study. It is also illustrated through an application on a real gene expression RNA-seq dataset in Section 8.5. After a short discussion, all the proofs are gathered in Section 8.7.

8.2 Detecting the block-diagonal structure by model selection

Let $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$ be iid random vectors in $\mathbb{R}^p$ from a multivariate distribution with density $\phi_p(0, \mathbf{\Sigma})$ where $\mathbf{\Sigma}_{j,j} = 1$ for all $j \in \{1, \dots, p\}$. The matrix $\mathbf{\Sigma}$ is assumed to have a block diagonal structure for some ordering of the variables. We note K the number of blocks, $B = (B_1, \dots, B_K)$ the partition of variables into K blocks, B_k the subset of variables in block k, p_k the number of variables in block k. The matrix $\mathbf{\Sigma}$ can be written as a block diagonal matrix with $\mathbf{\Sigma}^k \in \mathbb{S}_{p_k}^{++}(\mathbb{R})$ for $k \in \{1, \dots, K\}$:

$$\mathbf{\Sigma}_B = \begin{pmatrix} \mathbf{\Sigma}^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{\Sigma}^K \end{pmatrix}. \tag{8.2}$$

We define $\mathcal{B}$ the set of all possible partitions of variables. This set is large : there are $\sum_{k=1}^{p} S(p,k)$ possible partitions where $S(p,k)$ denotes the Stirling number of the second kind. Considering each element in the set $\mathcal{B}$ would not be possible. Following the *block diagonal screening rule* described in Mazumder and Hastie (2012), we consider the grid of threshold parameters $\Lambda = \left\{ (\hat{\mathbf{\Sigma}}_{j,j'})_{j>j'} \right\}$ derived from the empirical covariance matrix S. We restrict our attention to the sub-collection $\mathcal{B}^\Lambda$ of $\mathcal{B}$:

$$\mathcal{B}^\Lambda = \left\{ B(\lambda), \lambda \in \Lambda \text{ where } B(\lambda) \text{ are the connected components from } E^{(\lambda)} = (\mathbf{1}_{\{|S_{j,j'}|>\lambda\}})_{jj'} \right\}. \tag{8.3}$$

We define the model collection $\mathcal{F} = (F_B)_{B \in \mathcal{B}}$ where F_B is defined as follows for every $B \in \mathcal{B}$:

$$F_B = \{ f_B \text{ with } \mathbf{\Sigma}_B \in S_B \} \tag{8.4}$$

$$S_B = \left\{ \mathbf{\Sigma}_B \in \mathbb{S}_p^{++}(\mathbb{R}) \,\middle|\, \mathbf{\Sigma}_B = P_\sigma \begin{pmatrix} \mathbf{\Sigma}^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{\Sigma}^K \end{pmatrix} P_\sigma^{-1}, P_\sigma \text{ a permutation matrix}, \right.$$
$$\left. \mathbf{\Sigma}^k \in \mathbb{S}_{p_k}^{++}(\mathbb{R}) \text{ for } k \in \{1, \dots, K\} \right\}. \tag{8.5}$$

The dimension of the model F_B is $D_B = \sum_{k=1}^{K} \frac{p_k(p_k-1)}{2}$. The F_B contains all densities in the form $\phi(0, \mathbf{\Sigma}_B)$. For a given partition B, we define the maximum likelihood

estimator $\hat{f}_B$ as follows :

$$\hat{f}_B = \phi(0, \hat{\mathbf{\Sigma}}_B) \text{ with } \hat{\mathbf{\Sigma}}_B = \begin{pmatrix} \hat{\mathbf{\Sigma}}^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \hat{\mathbf{\Sigma}}^K \end{pmatrix} \text{ and } \hat{\mathbf{\Sigma}}^k = S^k \text{ for } k \in \{1, \ldots, K\}, \quad (8.6)$$

where S^k is the maximum likelihood estimator of the covariance matrix for the dataset restricted to the block k. We select the best partition using the following criterion :

$$\hat{B} = \underset{B \in \mathcal{B}^\Lambda}{\operatorname{argmin}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log(\hat{f}_B(\mathbf{y}_i)) + \text{pen}(B) \right\}, \quad (8.7)$$

where $\frac{1}{n} \sum_{i=1}^{n} \log(\hat{f}_B(\mathbf{y}_i))$ is the log-likehood of the model and pen a penalty term to be defined.

A first approach to select B would be to consider the classical BIC criterion proposed by Schwarz (1978), which leads to choose the following penalty :

$$\text{pen}(B) = \frac{\log n}{2} D_B.$$

Since the Gaussian graphical model fulfills regularity conditions, the BIC criterion asymptotically select the true partition $\hat{B} = B^\star$ where $B^\star$ is the true partition of variables into blocks, *i.e.* $\mathbf{y}$ is an iid random vector from a multivariate distribution with parameters $\mathbf{0}$ and $\mathbf{\Sigma}_{B^\star}$ (Haughton, 1988).

In our context, the number of observations n cannot be considered to be as large. For this reason, we consider a non-asymptotic model selection based on the slope heuristic, developed by Birgé and Massart (2007). This heuristic leads to the following penalty :

$$\text{pen}(B) = \kappa D_B, \quad (8.8)$$

where κ is a coefficient to calibrate. The model selected using this coefficient satisfies an oracle inequality. Theoretical guarantees are described in the following Section 8.3. Although the coefficient κ can be computed based on theory, we prefer to calibrate it from the data, using one of the two methods described in Baudry et al. (2012). One method is the *Slope Heuristic Dimension Jump* (SHDJ) : we approximate the coefficient κ by twice the coefficient corresponding to the biggest dimension jump on the graph representing the model dimension as a function of the coefficient κ. Another method is the *Slope Heuristic Robust Regression* (SHRR) : we approximate the coefficient κ by twice the slope of a robust regression between the log-likehood and the model dimension. The two methods are derived from the same heuristic and give similar results. They are implemented in the R package `capushe` (Baudry et al., 2012).

8.3 Theoretical results for non-asymptotic model selection

Model selection based on the slope heuristic in conjunction with the calibration of the κ coefficients by dimension jump (SHDJ) or robust regression (SHRR) have been proven

to be effective in a variety of practical situations. For example, Rau et al. (2015) select the number of components in Poisson mixture models on RNA-seq gene expression data using the slope heuristic. Bouveyron et al. (2015) select the number of components in discriminative functional mixture models on data describing bike sharing systems using the slope heuristic. However, they did not provide any theoretical justification for their procedures.

In contrast, we do provide theoretical justification for our criterion based on an oracle inequality. Lebarbier (2005) have provided theoretical justification based on an oracle inequality for model selection in multiple change point detection, and Maugis and Michel (2011) for variable selection in mixture models. However, few papers provide a minimax lower bound, which we do have. Remark that the theoretical justification of the slope heuristic has encountered several technical difficulties. The existence of minimal penalties is proved in heteroscedastic regression with fixed design (Birgé and Massart, 2007; Baraud et al., 2009), and for homoscedastic regression with fixed design (Arlot and Massart, 2009).

For our block-diagonal structure detection procedure, we prove an oracle inequality for a penalty proportional to the dimension (up to a logarithm term) and a lower bound of the risk between the true model and the model selected among the model collection. This ensures that the selected model is close to the oracle, the best one in estimation among our collection. Both inequalities guarantee that our model selection procedure has an optimal rate of convergence, which is a strong theoretical result. Note that these results are non-asymptotical, which means that they hold for a fixed sample size n.

To state the theorem, we recall the definition of the Hellinger distance between two densities f and g defined on $\mathbb{R}^p$,

$$d_H^2(f,g) = \frac{1}{2}\int_{\mathbb{R}^p} (\sqrt{f(x)} - \sqrt{g(x)})^2 dx = 1 - \int_{\mathbb{R}^p} \sqrt{f(x)g(x)}dx,$$

and the Kullback-Leibler divergence between two densities f and g defined on $\mathbb{R}^p$,

$$\mathrm{KL}(f,g) = \int_{\mathbb{R}^p} \log\left(\frac{f(x)}{g(x)}\right) f(x)dx.$$

In order to properly define the penalty term used in equation (8.8) to select the best partition of variables B, we work with the following model collection :

$$\mathcal{F}^{\text{bound}} = (F_B^{\text{bound}})_{B\in\mathcal{B}} \tag{8.9}$$

$$F_B^{\text{bound}} = \left\{\phi(0,\mathbf{\Sigma}_B) \in F_B, \mathbf{\Sigma}_B \in S_B^{\text{bound}}\right\} \tag{8.10}$$

$$\begin{aligned} S_B^{\text{bound}} = \{\mathbf{\Sigma}_B \in S_B | & e_m \leq \min(\mathbf{\Sigma}_B) \leq \max(\mathbf{\Sigma}_B) \leq e_M, \\ & \lambda_m \leq \Lambda_{\min}(\mathbf{\Sigma}_B) \leq \Lambda_{\max}(\mathbf{\Sigma}_B) \leq \lambda_M\}, \end{aligned}$$

where $\Lambda_{\min}(A)$ and $\Lambda_{\max}(A)$ are the smallest and the largest eigenvalues of the matrix A.

The model collection (8.9) is defined such that covariance matrices have bounded coefficients, which is useful for constructing a discretization of this space. If the matrix has bounded coefficients, we can prove that it has bounded eigenvalues. Nevertheless, to simplify the reading, we denote by λ_m and λ_M bounds on eigenvalues. In the following, we denote by $Adj(\mathbf{\Sigma})$ the adjacency matrix associated to the covariance matrix $\mathbf{\Sigma}$.

Theorem 8.3.1 *Let* $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ *be the observations, arising from a density* f^*. *Consider the model collection* $\mathcal{F}^{bound}$ *defined in* (8.9). *Suppose that there exists an absolute constant* $\kappa' > 0$ *such that for every partition* B *in the set of all possible partitions of variables* $\mathcal{B}$,

$$\text{pen}(B) \geq \kappa' \frac{D_B}{n} \left[2c^2 + \rho \log \left(\frac{1}{D_B(\frac{D_B}{n} c^2 \wedge 1)} \right) + (1 \vee \tau) \log \left(\frac{0.792p}{\log(p+1)} \right) \right],$$

where c *is an absolute constant, depending only on the model collection. Let* $\hat{f}_B$ *be the maximum likelihood estimator,* $\mathcal{B}^\Lambda \subset \mathcal{B}$ *as defined in* (8.3), *and* $\hat{B}$ *selected as follows :*

$$\hat{B} = \underset{B \in \mathcal{B}^\Lambda}{\text{argmin}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log(\hat{f}_B(\mathbf{y}_i)) + \text{pen}(B) \right\}.$$

Then, $\hat{f}_{\hat{B}}$ *satisfies :*

$$\mathbb{E}(d_H^2(f^*, \hat{f}_{\hat{B}})) \leq C\mathbb{E} \left(\inf_{B \in \mathcal{B}^\Lambda} \left(\inf_{t \in F_B^{bound}} KL(f^*, t) + \text{pen}(B) \right) + (1 \vee \tau) \frac{1}{n} \right) \tag{8.11}$$

for some absolute constant C.

This non-asymptotic result is consistent with the point of view adopted in this work where the number of observations n is limited. The proof is presented in Appendix 8.7.2. This theorem is deduced from an adaptation for a random sub-collection of the whole model collection of a general model selection theorem for maximum likelihood estimator developed by Massart (2007). This adaptation is proved in Appendix 8.7.2. To apply our theorem, the main assumptions to satisfy are the control of the bracketing entropy of each model in the whole model collection and the construction of weights for each model to control the model collection complexity. Remark that the control of the bracketing entropy is a classical tool to bound the Hellinger risk of the maximum likelihood estimator, and has already been done for Gaussian densities in Maugis and Michel (2011) and Genovese and Wasserman (2000).

Theorem 8.3.1 provides a lower bound for the penalty, which ensures a good model selection by penalized criterion : the model selected is as good as possible among the model collection. The only assumption made to state Theorem 8.3.1 is a classical one : we work with bounded parameters for each model as detailed in (8.9). Every constant involved in (8.11) depends on those bounds. Even if the bounds are not tractable in practice, this assumption is plausible. To guarantee a good model selection procedure, we need to assume that the true density of the data is not too far from the constructed model collection. Since a covariance matrix can always be considered to be a block-diagonal matrix, with possibly a single block, the block-diagonal covariance matrix assumption is not a strong one.

To complete this analysis, we provide a minimax lower bound for the risk between the true model and the model selected among the model collection. For the lower bound of the risk, some results have been previously obtained by Bickel and Levina (2008) and Cai et al. (2010). To obtain our lower bound, we use the lemma developed in Birgé (2005) in conjunction with a discretization of the model collection space, already constructed for the oracle inequality.

Theorem 8.3.2 *Let* $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ *be the observations, coming from a density* f^*. *Consider the model collection* $\mathcal{F}^{bound}$ *defined in* (8.9), *and* D_B *the dimension of the model* F_B^{bound} *for each* $B \in \mathcal{B}$. *Let* $\hat{f}_B$ *being the maximum likelihood estimator for the model indexed by* B. *Then, for all* $B \in \mathcal{B}$, *there exists absolute constants* $C_1 > 0$ *and* $C_2 > 0$ *such that :*

$$\inf_{\hat{f}_B} \sup_{f \in F_B^{bound}} \mathbb{E}(d_H^2(\hat{f}_B, f)) \geq C_1 \frac{D_B}{n}(1 + \log\left(\frac{C_2}{D_B^2}\right)). \tag{8.12}$$

This theorem is proved in Appendix 8.7.3. Again, this result does not rely on strong assumptions, and the constants involved are explicit. It is also a non-asymptotic result.

This minimax lower bound obviously shows that since the estimator satisfies to (8.11) it is simultaneously approximately minimax on each set F_B^{bound} for every $B \in \mathcal{B}$. Theorem 8.3.2 and Theorem 8.3.1 lead to the use of the slope heuristic with a penalty proportional to the dimension to select a model among the collection.

Nevertheless, as typically the case, constants are higher in theory than needed (and not always tractable), and we prefer to compute constants from the dataset in practice using the `capushe` package developed in Baudry et al. (2012).

8.4 Simulation study

8.4.1 Simulation setting

We simulate n observations from a $p-$multivariate normal distribution with a null mean and a block diagonal covariance matrix $\mathbf{\Sigma}_B$ as defined in (8.2). We fix the number of variables $p = 100$, the sample size $n = 90$ and the partition on variable $B^\star$: we vary the number of blocks among $K^\star \in \{1, 15\}$. For each block indexed by k, we design the $\mathbf{\Sigma}^k$ matrix as follows, as done in Giraud et al. (2012) : $\mathbf{\Sigma}^k = TT^T + D$ where T is a random lower triangular matrix with values drawn from a uniform distribution between -1 and 1, and D is a diagonal matrix designed to prevent $\mathbf{\Sigma}^k$ to have eigenvalues that are too small.

To perform network inference, we use the graphical lasso algorithm proposed in Friedman et al. (2008) and implemented in the R package `glasso`, version 1.7. We do not use the same thresholding parameter as in the block structure selection, and we denote by ρ the parameter of the graphical lasso algorithm for inferring the network. We compare the following strategies :

1. **Glasso :** graphical lasso on the set of all variables, with regularization parameter ρ chosen using the following BIC^{net} criterion :

$$\text{BIC}^{\text{net}}(\rho) = \frac{n}{2}\left(\log\det\hat{\mathbf{\Theta}}^{(\rho)} - \text{trace}\left(S\hat{\mathbf{\Theta}}^{(\rho)}\right)\right) - \frac{\log(n)}{2}\text{df}\hat{\mathbf{\Theta}}^{(\rho)}; \tag{8.13}$$

   where S is the sample covariance matrix, and df means the degree of freedom.

2. **CGL :** cluster graphical lasso as proposed in Tan et al. (2015) : first, the partition on variables is detected using an average linkage hierarchical clustering with

$K = 15$ clusters. Note that we set the number of clusters to the optimal number $K^\star$. Subsequently, the regularization parameters in each graphical lasso problem $\rho_1, \ldots, \rho_{K^\star}$ are chosen from the Corollary 3 from Tan et al. (2015) : the inferred network in each block must be as sparse as possible while still remaining a single connected component.

3. **Partitions based on model selection :** depending on the procedure, the first step is based on asymptotic (BIC) our non asymptotic model selection (SHRR ou SHDJ).

   (a) BIC : The partition $\hat{B}_{\text{BIC}}$ is selected using the BIC criterion (Schwarz, 1978).

   (b) SHRR : The partition $\hat{B}_{\text{SHRR}}$ is selected using the *Slope Heuristic Robust Regression.*

   (c) SHDJ : The partition $\hat{B}_{\text{SHDJ}}$ is detected using the *Slope Heuristic Dimension Jump.*

   Subsequently, the regularization parameters in each graphical lasso problem $\rho_1, \ldots, \rho_{\hat{K}}$ are chosen using the BIC^{net} criterion :

$$\text{BIC}^{\text{net}}(\rho_k) = \frac{n}{2}\left(\log\det\hat{\mathbf{\Theta}}^{(\rho_k)} - \text{trace}\left(S_{|k}\hat{\mathbf{\Theta}}^{(\rho_k)}\right)\right) - \frac{\log(n)}{2}\text{df}\hat{\mathbf{\Theta}}^{(\rho_k)}, \qquad (8.14)$$

   where $S_{|k}$ is the sample covariance matrix on variables belonging to the block k and df means the degree of freedom.

4. **True partition (truePart) :** first, we set the partition on variables to the true partition $B^\star$. Then, the regularization parameters in each graphical lasso problem $\rho_1, \ldots, \rho_{K^\star}$ are chosen using the BIC^{net} criterion (8.14).

We compare the performance of the five methods using the sensitivity (SENS), (SENS = TP/(TP + FN)), the specificity (SPEC) SPEC = TN/N = TN/(TN + FP) and the False Discovery Rate (FDR) (FDR = FP/(TP + FP)) where TN, TP, FN, FP are respectively the number of true negative, true positive, false negative, false positive dependencies detected. A network inference procedure is a compromise between sensitivity and specificity : we are looking for a high sensitivity, which measures the proportion of dependencies (presence of edges) that are correctly identified, and a high specificity, which measures the proportion of independencies (absence of edges) that are correctly identified. The accuracy is a global measure of quality of the network inference performance. The False Discovery Rate is the proportion of dependencies wrongly detected. As a rule of thumb, we expect that the FDR remains below 0.05 for common statistical analysis. We compute the Mean Squared Error (MSE) between the true and the estimated covariance matrices. We also investigate the ability to recover the simulated partition on variables $B^\star$ using the hierarchical clustering from Tan et al. (2015), the BIC, SHRR and SHDJ partition selection methods. Selected partitions for each method are compared with the simulated partition $B^\star$ using the Adjusted Rand Index (Hubert and Arabie, 1985).

8.4.2 Results

Block diagonal covariance matrix $\mathbf{\Sigma}$ with $K^\star = 15$ blocks

We simulate data from a multivariate normal distribution with a null mean and a block diagonal covariance matrix $\mathbf{\Sigma}$ with $K^\star = 15$ blocks of approximate equal sizes (6 or 7 variables). The performance of the different strategies is illustrated on Figure 8.1. As expected, the true partition strategy (truePart) performs the best : based on the true partition of variables, network inference problem is easier because we solve problems with smaller dimension. The proposed strategies, based on the BIC, SHRR and SHDJ partitions, improve network inference compared to a simple graphical lasso on the set of all variables (glasso) or compared to the cluster graphical lasso (CGL). Note that the strategy based on the BIC partition performs well. This result is not surprising because data has been simulated under the model. Illustrations of the calibration of coefficient κ are presented in Figures 8.2. In addition, we compare the partition selection methods with an average linkage hierarchical clustering with $K = 15$ as proposed in the cluster graphical lasso (Tan et al., 2015). The Figure 8.3 displays the ARI computed over 100 replicated datasets. Despite the fact that the partition with the hierarchical clustering has the true number of clusters ($K = 15$), the ARI for the hierarchical clustering is lower than the ARI for the three methods (BIC, SHRR and SHDJ) which do not need to specify the number of clusters K in advance, contrary to the hierarchical clustering. The partition of variables selected by BIC, SHRR and SHDJ agreed for $n = 90$. When n is smaller ($n = 30$), the robust regression (SHRR) and dimension jump (SHDJ) give slightly better results than the asymptotic criterion BIC.

Full covariance matrix $\mathbf{\Sigma}$ with $K^\star = 1$ blocks

We simulate $n = 90$ observations from a multivariate normal distribution with a null mean and full covariance matrix $\mathbf{\Sigma}$. The corresponding network of conditional dependencies is almost a clique. As we suspected, solving the graphical lasso problem in this context is too ambitious as proved by (Verzelen, 2012) : detecting the true network requires the estimation of $D = 4950$ parameters, with only $n \times p = 900$ data points. The log-likelihood of the model displayed in Figure 8.4. In contrast with the Figure 8.2, no linear tendency is observed between the model dimension and the log-likelihood function for complex models. In this context, no relevant block diagonal structure is detected.

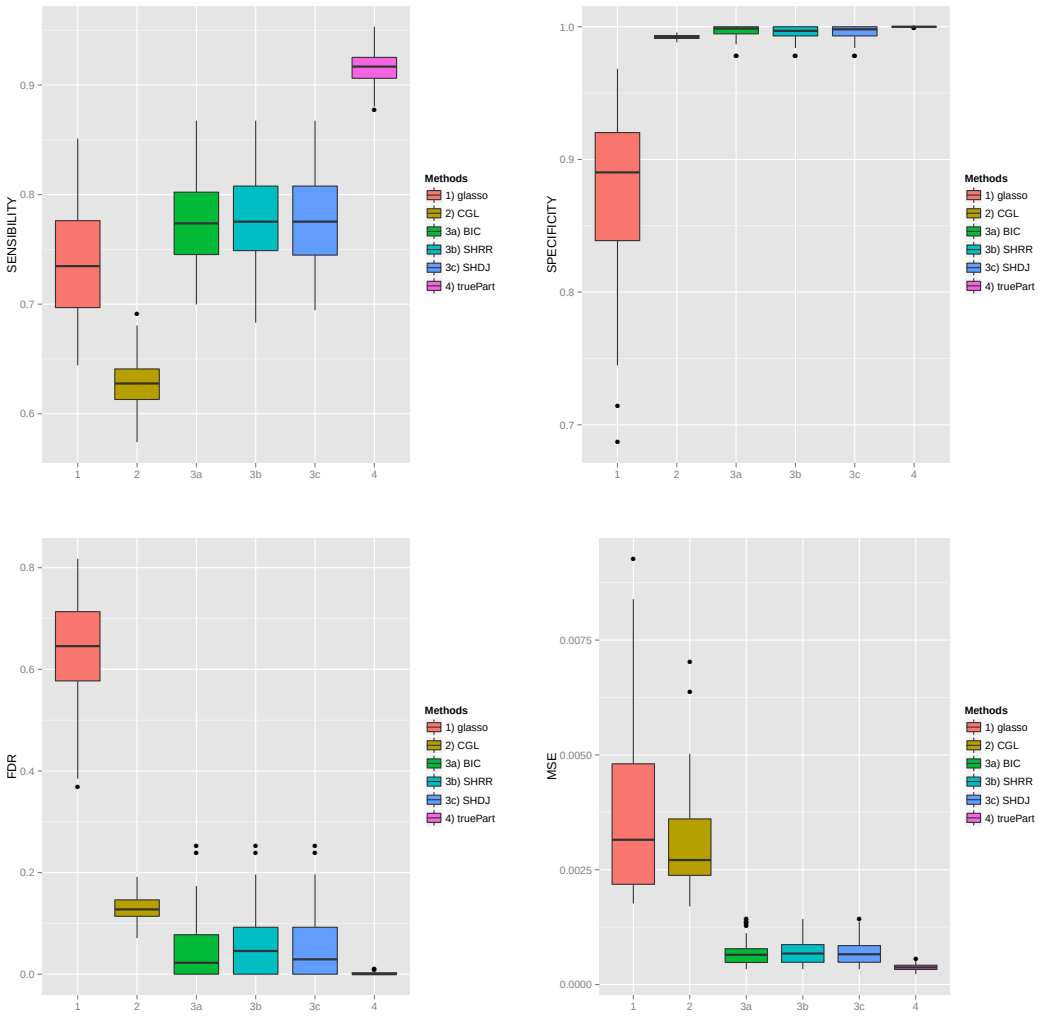

FIGURE 8.1 – Performance of network inference methods (glasso : graphical lasso on the set of all variables, CGL : cluster graphical lasso, BIC : network inference based on the partition of variables $\hat{B}_{\text{BIC}}$, SSHR : network inference based on the partition of variables $\hat{B}_{\text{SHRR}}$, SHDJ : network inference based on the partition of variables $\hat{B}_{\text{SHDJ}}$ and truePart : network inference based on the partition of variables $B^\star$) measured by the sensitivity (SENS), the specificity (SPEC), the False Discovery Rate (FDR) of the inferred graph, and the Mean Squared Error (MSE) between the true and estimated covariance matrices over 100 replicated datasets simulated under a $p-$multivariable normal distribution with a null mean $\mathbf{0}$ and a block diagonal covariance matrix $\mathbf{\Sigma}_{B^\star}$ with $p = 100$, $K = 15$, $n = 90$ and clusters of approximate equal sizes.

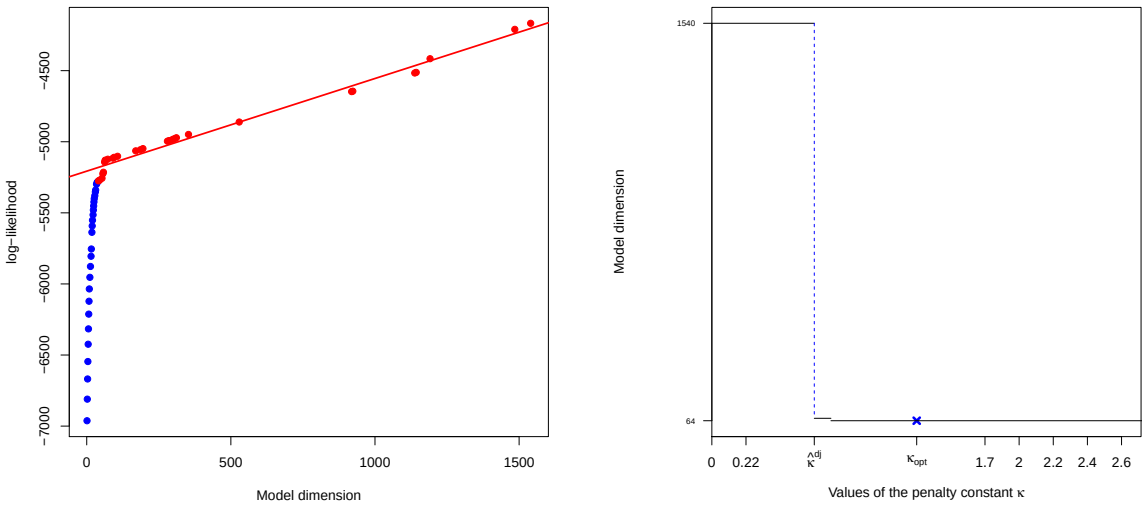

FIGURE 8.2 – Calibration of the κ coefficient on a dataset simulated under a multivariate normal distribution with a block diagonal covariance matrix $\mathbf{\Sigma}_B$ with $K^\star = 15$ blocks, $p = 100$, $n = 90$. Calibration by robust regression (left) : the log-likehood of the model is represented as a function of the model dimension. Based on the slope heuristic, the slope of the regression (red line) between the log-likehood and the model dimension for complex models (red points) corresponds to the minimal coefficient κ_{min}. The optimal penalty is twice the minimal penalty. Calibration by dimension jump (right) : the dimension of the model is represented as a function of the κ coefficient. Based on the slope heuristic, the biggest jump (dotted blue line) corresponds to the minimal coefficient κ_{min}. The optimal penalty (blue cross) is twice the minimal penalty.

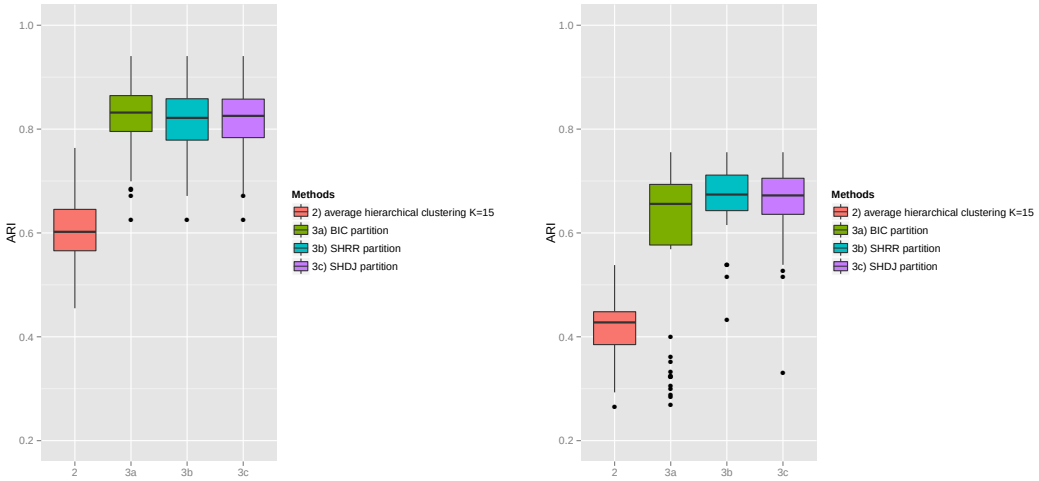

FIGURE 8.3 – ARI between the simulated partition and the partitions selected with BIC, the robust regression (SHRR), dimension jump (SHDJ) and by average hierarchical clustering with $K = 15$ clusters. The ARI are computed over 100 replicated datasets simulated under a multivariate normal distribution ($p = 100$) with block diagonal covariance matrix ($K = 15$) with $n = 90$ (left) and $n = 30$ observations (right).

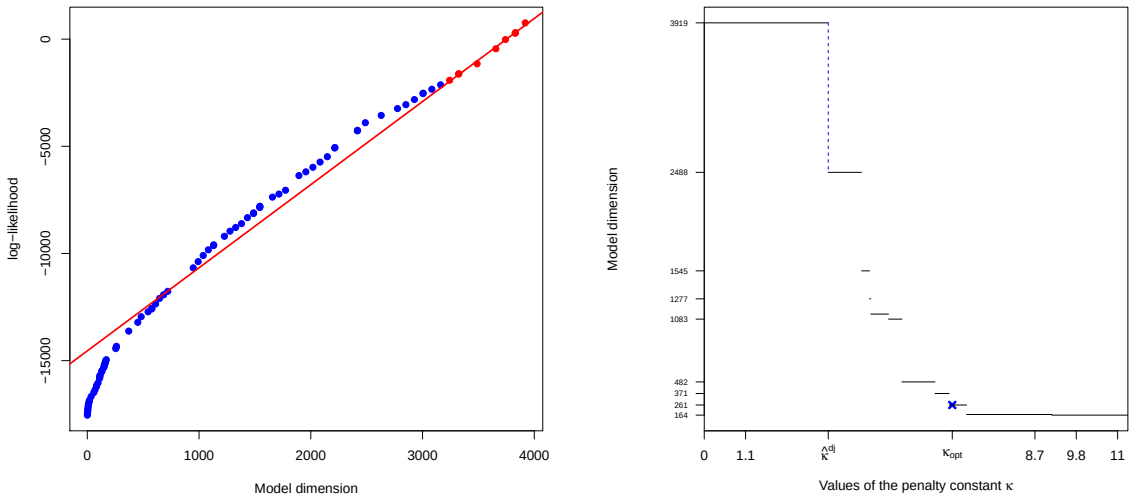

FIGURE 8.4 – Calibration of the κ coefficient on a dataset simulated under a multivariate normal distribution with a full covariance matrix with one $K^\star = 1$ and $p = 100$, $n = 90$. Calibration by robust regression (left) and by dimension jump (right) : in this extreme setting, no clear linear tendency (red line) between the log-likelihood and the model complexity for complex models (red points) is observed, the biggest largest jump (dotted blue line) is unclear.

8.5 Real data analysis

Pickrell *et al.* analyzed transcriptome expression variation from 69 lymphoblastoid cell lines derived from unrelated Nigerian individuals (Pickrell, 2010). The expression of 52580 genes across 69 observations was measured using RNA-seq. The data is extracted from the Recount database (Frazee et al., 2011). First, we filter weakly expressed genes using the `HTSFilter` package (Rau et al., 2013). Among the 9191 remaining genes, we identify the 200 most variable genes and restrict our attention to this set of genes for the following network inference analysis.

First, we select the partition $\hat{B}$ using model selection as described in equation (8.7). The log-likelihood increases with the number of parameters to be estimated in the model as displayed in Figure (8.5). We notice a linear tendency in the relationship between the log-likelihood and the model dimension for complex models (points corresponding to a model dimension higher than 500). This suggests that the use of the slope heuristic is appropriate for selecting a partition $\hat{B}$. The model selected by *slope heuristic robust regression* and by *slope heuristic dimension jump* described in section 8.2 are the same. The number of blocks detected is $\hat{K}_{\text{SH}} = 150$ and the corresponding model dimension is $D_{\hat{B}_{\text{SH}}} = 283$. The partition $\hat{B}_{\text{SH}}$ yields 4 blocks of size $18, 13, 8$ and 5, 4 blocks of size 3, 2 blocks of size 2 and 140 blocks of size 1. The partition selected by the Slope Heuristic offers a drastic reduction of the number of parameters to infer, as compared with the graphical lasso performed on the full set of variables which corresponds to a total of $D = 19900$ parameters to estimate. We also compare the partition $\hat{B}_{\text{SH}}$ selected by the slope heuristic with that selected by BIC $\hat{B}_{\text{BIC}}$: $\hat{K}_{\text{BIC}} = 99$ and $D_{\hat{B}_{\text{BIC}}} = 2213$ with 6

blocks of size $65, 10, 8, 8, 7$ and 4, 5 clusters of size 2 and 88 clusters of size 1.

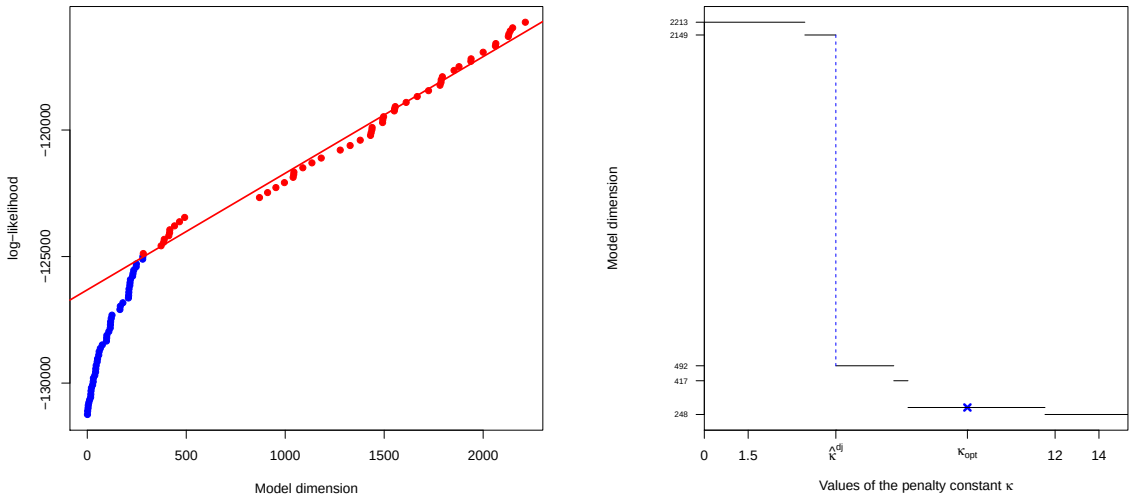

FIGURE 8.5 – Calibration of the κ coefficient on the 200 most variable genes extracted from the (Pickrell, 2010) dataset. Calibration by robust regression (left) : the log-likehood of the model is represented as a function of the model dimension. Based on the slope heuristic, the slope of the regression (red line) between the log-likehood and the model dimension for complex models corresponds to the minimal coefficient κ_{min}. The optimal penalty is twice the minimal penalty. Calibration by dimension jump (right) : the dimension of the model is represented as a function of the κ coefficient. Based on the slope heuristic, the biggest jump corresponds to the minimal coefficient κ_{min}. In both cases, the optimal penalty is twice the minimal penalty.

The networks within each cluster of variables are inferred using the graphical lasso algorithm of Friedman (Friedman et al., 2008) implemented in the `glasso` package, version 1.7. The regularization parameter for the graphical lasso on the set of all variables is chosen using the BIC^{net} criterion (8.13). The model inferred based on partition $\hat{B}_{\text{SH}}$ is more parsimonious and easier to interpret than the model inferred on the full set of variables or the model based on the partition $\hat{B}_{BIC}$. An illustration of inferred networks in the four largest connected components of the partition $\hat{B}_{\text{SH}}$ are displayed on Figure 8.6. These three networks might be good candidates for further study.

8.6 Discussion

In this chapter, we propose a non-asymptotic procedure to detect block diagonal structure for covariance matrices in Gaussian graphical models. Subsequently, we infer the network in each block using the graphical lasso algorithm. The procedure substantially reduces the number of parameters to estimate in the model. Although Gaussian graphical models are widely used in practice, the limited sample size forces the user to restrict the number of variables to include in the model. Usually, this restriction is performed manually, for instance, based on prior knowledge on the role of variables.

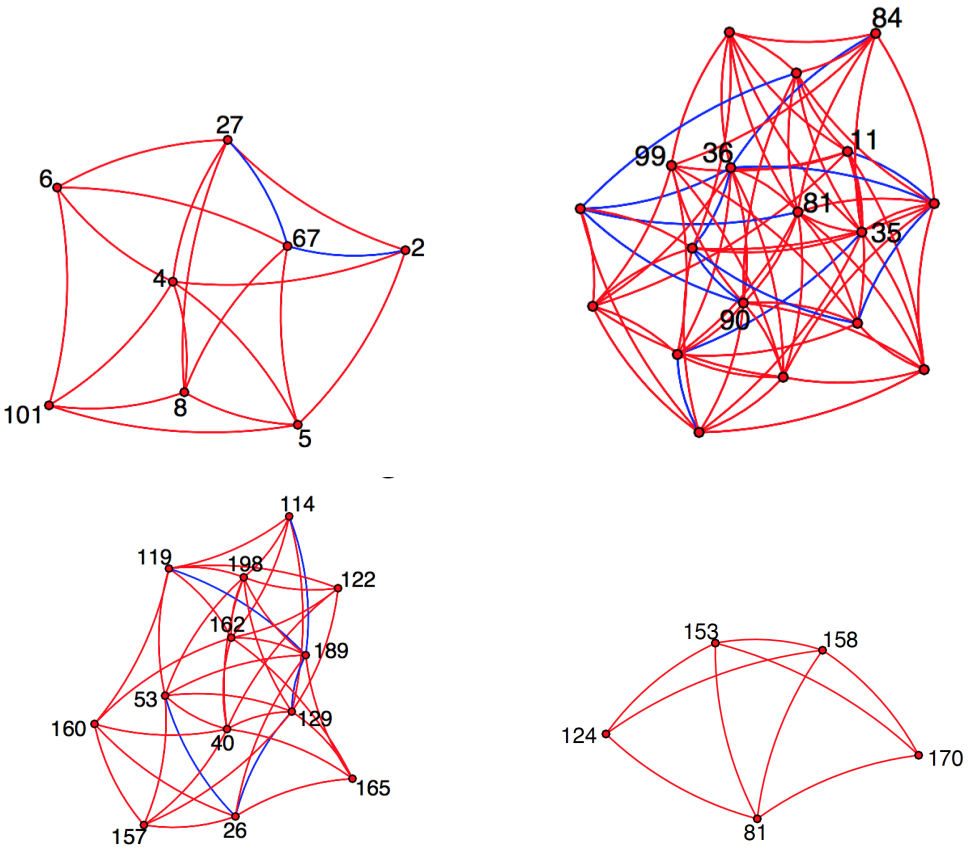

FIGURE 8.6 – Networks inferred on the four largest components detected by slope heuristic. Regularization parameters in each set of variables are chosen using the $\mathrm{BIC}^{\mathrm{net}}$ criterion (8.14). Numbers indicate genes labels. Red edges indicate positive partial correlation whereas blue edges indicates negative partial correlation between genes.

Here, we propose a procedure to select relevant subsets of variables based on the data. Therefore, our procedure is of great practical interest to estimate parameters in Gaussian graphical models when the sample size is smaller than the number of parameters to estimate.

The procedure we propose is easy to implement in practice and fast to compute. The calibration of the κ coefficient by robust regression and dimension jump encounter no particular difficulty. Moreover, graphical representation of the log-likelihood or the model dimension can indicate if the block diagonal assumption is wrong, for instance in extreme cases where the covariance matrix is not sparse and no block diagonal structure can be detected.

Our method uses a model selection criterion to detect block diagonal structure. We propose a non-asymptotic criterion supported by strong theoretical results. It is important to highlight that, although BIC can be very efficient when the model is appropriate, it can provide unsatisfactory results in more difficult inference situations, for example, in data with large bias (as is often seen in real data). In such cases, the slope heuristic appears to provide satisfactory results and yield more interpretable models, especially when the model family is somewhat biased with respect to the sampling distribution. The BIC criterion does not deal with the fact that the block diagonal structure assumption is an approximation of the reality, not the truth. The BIC criterion performs well on simulated data because the data has been simulated under the model, with a block diagonal covariance matrix. By taking into account both the model complexity and the size of the model collection considered into the penalty term, the non-asymptotic model selection provides a parsimonious model choice even in situations where the BIC tends to overestimate the number of parameters.

8.7 Appendix

In this Appendix, we detail the proof of Theorems 8.3.1 and 8.3.2. First, we describe a discretization of the model collection used, which is useful in the two proofs. Then, in Section 8.7.2, we prove Theorem 8.3.1. We first generalize a model selection theorem for MLE, introduced by Massart, to random model selection. Subsequently, we prove that our model collection satisfies all the assumptions of this Theorem, and deduce the oracle inequality. In Section 8.7.3, we prove Theorem 8.3.2 using Birgé's Lemma (lemma 8.7.4) with the discretization of the model collection obtained in Section 8.7.1.

8.7.1 Model collection and discretization

Discretization for the adjacency matrices

Let $B = (B_1, \ldots, B_K) \in \mathcal{B}$. For a given matrix $\mathbf{\Sigma}_B \in S_B^{\text{bound}}$, we may identify a corresponding adjacency matrix A_B. This matrix of size p^2 could be summarized by the vector of concatenated upper triangular vectors. Then, we construct a discrete space for $\{0,1\}^{p(p-1)/2}$ which is in bijection with :

$$\mathcal{A}_B^{\text{bound}} = \left\{ A_B \in \mathbb{S}_p(\{0,1\}) | \exists \mathbf{\Sigma}_B \in S_B^{\text{bound}} \text{ s.t. } Adj(\mathbf{\Sigma}_B) = A_B \right\}.$$

First, we focus on the set $\{0,1\}^{p(p-1)/2}$.

Lemma 8.7.1 *Let $\{0,1\}^{p(p-1)/2}$ be equipped with Hamming distance δ. Let $\{0,1\}_B^{p(p-1)/2}$ be the subset of $\{0,1\}^{p(p-1)/2}$ of vectors for which the corresponding graph has structure B.*

For every $\alpha \in (0,1)$, let $\beta \in (0,1)$ such that $D_B \leq \alpha\beta p(p-1)/2$. There exists some subset $\mathcal{R}(\alpha)$ of $\{0,1\}_B^{p(p-1)/2}$ with the following properties

$$\delta(r, r') > 2(1-\alpha)D_B \text{ for every } (r, r') \in \mathcal{R}(\alpha)^2 \text{ with } r \neq r' \tag{8.15}$$

$$\log|\mathcal{R}(\alpha)| \geq \rho D_B \log\frac{p(p-1)}{2D_B} + \kappa K(1-\log(K)) \tag{8.16}$$

where $\rho = -\alpha(-\log(\beta)+\beta-1)/\log(\alpha\beta)$ and $D_B = \sum_{1\leq k\leq K} p_k(p_k-1)/2$.

Proof. Let $\mathcal{R}$ be a maximal subset of $\{0,1\}_B^{p(p-1)/2}$ satisfying property (8.15). Then the closed balls with radius ϵ whose belongs to $\mathcal{R}$ cover $\{0,1\}_B^{p(p-1)/2}$. We remark that $x \mapsto P_\sigma x P_\sigma^{-1}$ is a group action, isometric and transitive on $\{0,1\}_B^{p(p-1)/2}$.

Hence,

$$|\{0,1\}_B^{p(p-1)/2}| \leq \sum_{x\in\mathcal{R}} |B_{\{0,1\}_B^{p(p-1)/2}}(x,\epsilon)| = |\mathcal{R}||B_{\{0,1\}_B^{p(p-1)/2}}(x^0,\epsilon)|$$

for every $x^0 \in \mathcal{R}$, where $B_A(x,r) = \{y \in A | \delta(x,y) \leq r\}$.

Our proof is similar to the proof of Lemma 4.10 in Massart (2007). We consider :

$$[\{0,1\}^{p(p-1)/2}]_D = \left\{x \in \{0,1\}^{p(p-1)/2} | \delta(0,x) = D\right\}.$$

Let $\alpha \in (0,1), \beta \in (0,1)$ such that $D \leq \alpha\beta p(p-1)/2$. According to Massart (2007), we know that :

$$|B_{[\{0,1\}^{p(p-1)/2}]_D}(x^0, 2(1-\alpha)D)| \leq \frac{\exp\left(-\rho D\log(p(p-1)/2D)\right)}{\binom{p(p-1)/2}{D}}.$$

with $\rho = -\alpha(-\log(\beta)+\beta-1)/\log(\alpha\beta)$.

Nevertheless, as $\{0,1\}_B^{p(p-1)/2} \subset [\{0,1\}^{p(p-1)/2}]_{D_B}$, for $D_B = \sum_{k=1}^K p_k(p_k-1)/2$,

$$|\{0,1\}_B^{p(p-1)/2}| \leq |\mathcal{R}|\frac{\exp\left(-\rho D_B\log(p(p-1)/2D_B)\right)}{\binom{p(p-1)/2}{D_B}}.$$

As $\{0,1\}_B^{p(p-1)/2}$ corresponds to the stabilizer of x^0,

$$|\{0,1\}_B^{p(p-1)/2}| \geq \frac{p!}{p_1!\ldots p_K!K!}.$$

Note that we divide by $K!$ because there are at worst K clusters with the same size.

As

$$\frac{p!}{p_1!\dots p_K!} \geq 1 \qquad \text{and} \qquad \binom{p(p-1)/2}{D_B} \geq 1,$$

$$|\mathcal{R}| \geq \frac{1}{K!} \exp\left(\rho D_B \log(p(p-1)/2D_B)\right).$$

Using Stirling's approximation, we obtain :

$$\log(|\mathcal{R}|) \geq \kappa K(1-\log(K)) + \rho D_B \log\left(\frac{p(p-1)}{2D_B}\right).$$

Discretization for the set of covariance matrices

Let $\alpha \in (0,1)$ and $\beta \in (0,1)$ such that $D_B \leq \alpha\beta p(p-1)/2$. Let $\mathcal{R}(\alpha)$ as constructed in Lemma 8.7.1, and its equivalent $\mathcal{A}_B^{\text{disc}}(\alpha)$ for adjacency matrices. Let $\epsilon > 0$. Let :

$$S_B^{\text{disc}}(\epsilon,\alpha) = \left\{\boldsymbol{\Sigma} \in \mathbb{S}_p^{++}(\mathbb{R}) | \text{Adj}(\boldsymbol{\Sigma}) \in \mathcal{A}_B^{\text{disc}}(\alpha), \boldsymbol{\Sigma}_{i,j} = \sigma_{i,j}\epsilon, \sigma_{i,j} \in \left[\frac{e_m}{\epsilon}, \frac{e_M}{\epsilon}\right] \cap \mathbb{Z}\right\}.$$

Then,

$$||\boldsymbol{\Sigma} - \boldsymbol{\Sigma}'||_2^2 \geq 2(1-\alpha)D_B \wedge \epsilon \text{ for every } (\boldsymbol{\Sigma}, \boldsymbol{\Sigma}') \in (S_B^{\text{disc}}(\epsilon,\alpha))^2 \text{ with } \boldsymbol{\Sigma} \neq \boldsymbol{\Sigma}'$$
$$\log|S_B^{\text{disc}}(\epsilon,\alpha)| \geq \rho D_B \log\left(\left\lfloor \frac{e_M - e_m}{\epsilon} \right\rfloor \frac{p(p-1)}{2D_B}\right) + \kappa K(1-\log(K)).$$

Proof. Let $(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}') \in (S_B^{\text{disc}}(\epsilon,\alpha))^2$ with $\boldsymbol{\Sigma} \neq \boldsymbol{\Sigma}'$. If $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'$ are close, either they have the same adjacency matrix and they differ only on a coefficient or they differ in their adjacency matrices. In the first case, $||\boldsymbol{\Sigma} - \boldsymbol{\Sigma}'||_2^2 \geq \epsilon$. In the second case, $||\boldsymbol{\Sigma} - \boldsymbol{\Sigma}'||_2^2 \geq 2(1-\alpha)D_B$. Then,

$$||\boldsymbol{\Sigma} - \boldsymbol{\Sigma}'||_2^2 \geq 2(1-\alpha)D_B \wedge \epsilon,$$

this minimum depending on α and ϵ.

8.7.2 Oracle inequality : proof of Theorem 8.3.1

First, we state the general theorem we use to get the oracle inequality, and its proof. Then, we deduce the oracle inequality by proving that our model collection satisfies all the assumptions.

Model selection theorem for MLE among a random sub-collection

We denote by $\mathcal{H}_{[.]}(\epsilon, S, d_H)$ the bracketing entropy of the set S with ϵ-brackets according to the Hellinger distance d_H.

Theorem 8.7.2 *Let f^* be an unknown density to be estimated from a sample of size n $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$. Consider $\{F_m\}_{m\in\mathcal{M}}$ some at most countable deterministic model collection. Let $\{w_m\}_{m\in\mathcal{M}}$ be some family of nonnegative numbers such that :*

$$\sum_{m\in\mathcal{M}} \exp(-w_m) \leq \Omega < \infty. \tag{8.17}$$

We assume that for every $m \in \mathcal{M}$, $\sqrt{\mathcal{H}_{[.]}(\epsilon, F_m, d_H)}$ is integrable in 0.

Moreover, for all $m \in \mathcal{M}$, we assume that there exists ψ_m on $\mathbb{R}_+$ such that ψ_m is nondecreasing, $\xi \mapsto \psi_m(\xi)/\xi$ is non-increasing on $(0, +\infty)$, and for all $\xi \in \mathbb{R}^+$, for all $u \in F_m$, denoting by $F_m(u, \xi) = \{t \in F_m, d_H(t, u) \leq \xi\}$,

$$\int_0^\xi \sqrt{\mathcal{H}_{[.]}(\epsilon, F_m(u, \xi), d_H)} d\epsilon \leq \psi_m(\xi). \tag{8.18}$$

Let ξ_m such that $\psi_m(\xi_m) = \sqrt{n}\xi_m^2$.

Introduce $\{F_m\}_{m\in\tilde{\mathcal{M}}}$ some random sub-collection of $\{F_m\}_{m\in\mathcal{M}}$. Let $\tau > 0$, and for all $m \in \mathcal{M}$, let $f_m \in F_m$ such that :

$$KL(f^*, f_m) \leq 2 \inf_{t\in F_m} KL(f^*, t);$$
$$f_m \geq \exp(-\tau) f^*. \tag{8.19}$$

Let $\eta \geq 0$ and consider the collection of η-maximum likelihood estimators $\{\hat{f}_m\}_{m\in\tilde{\mathcal{M}}}$. Let pen : $\mathcal{M} \to \mathbb{R}^+$. Suppose that there exists an absolute constant $\kappa > 0$ such that for all $m \in \mathcal{M}$:

$$\text{pen}(m) \geq \kappa \left(\xi_m^2 + (1 \vee \tau) w_m/n\right).$$

Let $\eta' \geq 0$. Then, $\hat{f}_{\hat{m}}$, with $\hat{m} \in \tilde{\mathcal{M}}$ such that :

$$-\frac{1}{n}\sum_{i=1}^n \log(\hat{f}_{\hat{m}}(\mathbf{y}_i)) + \text{pen}(\hat{m}) \leq \inf_{m\in\tilde{\mathcal{M}}} \left\{ -\frac{1}{n}\sum_{i=1}^n \log(\hat{f}_m(\mathbf{y}_i)) + \text{pen}(m) \right\} + \eta'$$

satisfies :

$$\mathbb{E}(d_H^2(f, \hat{f}_{\hat{m}})) \leq C \left(\inf_{m\in\tilde{\mathcal{M}}} \inf_{t\in F_m} KL(f, t) + \text{pen}(m) \right) + (1 \vee \tau)\frac{\Omega^2}{n} + \eta + \eta',$$

for some absolute positive constant C.

This theorem is a generalization of Theorem 7.11 in Massart (2007) to a random model subcollection of the whole collection. As the proof is adapted from the proof of this theorem, we detail here only differences and we refer the interested reader to Massart (2007).

We denote by γ_n the empirical process and by $\bar{\gamma}_n$ the centered empirical process. Following the proof of the Massart's theorem, easy computations lead to :

$$2KL\left(f, \frac{f + \hat{f}_{m'}}{2}\right) \leq KL(f, f_m) + \text{pen}(m) - \text{pen}(m') + 2(\bar{\gamma}_n(g_m) - \bar{\gamma}_n(\hat{s}_{m'}))$$

where

$$g_m = -\frac{1}{2}\log\left(\frac{f_m}{f}\right) \qquad \text{and} \qquad \hat{s}_m = -\log\left(\frac{f+\hat{f}_m}{2f}\right)$$

for $m \in \tilde{\mathcal{M}}$ and $m' \in \tilde{\mathcal{M}}(m) = \left\{m' \in \tilde{\mathcal{M}}, \gamma_n(\hat{f}_{m'}) + \text{pen}(m') \leq \gamma_n(\hat{f}_m) + \text{pen}(m)\right\}$.

To bound $\bar{\gamma}_n(\hat{s}_{m'})$, we use Massart's arguments. The main difference stands in the control of $\bar{\gamma}_n(g_m)$. As $\tilde{\mathcal{M}} \subset \mathcal{M}$ is random, $\mathbb{E}(\bar{\gamma}_n(g_m)) \neq 0$. Nevertheless, thanks to the Bernstein inequality, which we may use thanks to the inequality in (8.19), we obtain, for all $u > 0$, with probability smaller than $\exp(-u)$,

$$\nu_n(g_m) \leq \sqrt{\frac{1}{n}\alpha_\tau(1 \vee \tau)KL(f, f_m)u} + \frac{\tau}{2n}u,$$

where α_τ is a constant depending on τ. Then, choosing $u = w_m$ for all $m \in \mathcal{M}$, where w_m is defined in (8.17), some fastidious but straightforward computations similar to those of Massart's lead to Theorem 8.7.2.

We remark that this is a theoretically easy extension, but quite useful in practice, *e.g.* for controlling large model collections.

Bracketing entropy

Let $B \in \mathcal{B}$. Let $f \in F_B^{\text{bound}} : f = \Phi(0, \mathbf{\Sigma}_B)$. Let $\epsilon > 0$ and $\alpha > 0$. According to Corollary 8.7.1, there exists $S \in S_B^{\text{disc}}(\epsilon, \alpha)$ such that :

$$||\mathbf{\Sigma}_B - S||_2^2 \leq 2(1-\alpha)D_B \wedge \epsilon.$$

If we take $\alpha = 1 - \epsilon/2D_B$, we obtain $||\mathbf{\Sigma}_B - S||_2^2 \leq \epsilon$.

Then we consider :

$$u(x) = (1+2\delta)^\gamma \phi(x|0, (1+\delta)S),$$
$$l(x) = (1+2\delta)^{-\gamma} \phi(x|0, (1+\delta)^{-1}S).$$

According to the Proposition 4 in Maugis and Michel (2011), if $\delta = \beta/\sqrt{3}\gamma$ and if $\epsilon = \lambda_m \beta/(3\sqrt{3}p^2)$, the set $\{l, u\}$ is a β-bracket set over F_B^{bound}.

If we denote by $\mathcal{N}_{[.]}(\beta, F_B^{\text{bound}}, d_H)$ the minimal number of brackets $[l, u]$ such that $d_h(l, u) \leq \epsilon$ which are necessary to recover F_B^{bound} and $\mathcal{H}_{[.]}(\beta, F_B^{\text{bound}}, d_H)$ the logarithm of this number, which corresponds to the bracketing entropy, we obtain from Corollary 8.7.1 that :

$$\mathcal{N}_{[.]}(\beta, F_B^{\text{bound}}, d_H) \leq \kappa \left(\frac{3\sqrt{3}p^2(e_M - e_m)p(p-1)}{\lambda_m 2D_B\beta}\right)^{\rho D_B} K(1 - \log(K))$$
$$\mathcal{H}_{[.]}(\beta, F_B^{\text{bound}}, d_H) \leq D_B\left(\log C + \rho \log\left(\frac{1}{D_B\epsilon}\right)\right).$$

with $C = \kappa K(1 - \log(K))^{\frac{3\sqrt{3}p^2(e_M - e_m)p(p-1)}{2\lambda_m}}$.

We then construct ψ_B satisfying Equation (8.18).

For all $\xi > 0$,

$$\int_0^\xi \sqrt{\mathcal{H}_{[.]}(\beta, F_B^{\text{bound}}, d_H)}d\beta \leq \xi\sqrt{D_B \log C} + \sqrt{D_B\rho}\int_0^\xi \sqrt{\log\left(\frac{1}{D_B\beta}\right)}d\beta.$$

According to Maugis and Michel (2011),

$$\int_0^\xi \sqrt{\log\left(\frac{1}{\beta}\right)}d\beta \leq \int_0^{\xi\wedge 1} \sqrt{\log\left(\frac{1}{\beta}\right)}d\beta \leq (\xi\wedge 1)\left(\sqrt{\pi} + \sqrt{\log\left(\frac{1}{\xi\wedge 1}\right)}\right).$$

Then, denoting by $c = \sqrt{\log C} + \sqrt{\pi}$, we can define ψ_B by :

$$\psi_B(\xi) = \sqrt{D_B}\xi\left(c + \sqrt{\rho\log\frac{1}{D_B}} + \sqrt{\rho\log\frac{1}{\xi\wedge 1}}\right).$$

As we want ξ_B such that $\psi_B(\xi_B) = \sqrt{n}\xi_B^2$, we could take :

$$\xi_B^2 \leq \frac{D_B}{n}\left[2c^2 + \rho\log\left(\frac{1}{D_B(\frac{D_B}{n}c^2\wedge 1)}\right)\right].$$

Construction of the weights

We need to control the Bell number, which is the cardinal of $\mathcal{B}$. For this, we use a result of Berend and Tassa (2010), which guarantees the following inequality for $p \in \mathbb{N}$:

$$|\mathcal{B}| \leq \left(\frac{0.792p}{\log(p+1)}\right)^p.$$

For every $B \in \mathcal{B}$, we know that $p \leq D_B \leq p(p-1)/2$. Then, we obtain the following result.

Lemma 8.7.3 *Let* $w_B = D_B \log\left(\frac{0.792p}{\log(p+1)}\right)$. *Then,* $\sum_{B\in\mathcal{B}}\exp(-w_B) \leq 1$.

8.7.3 Lower bound for the minimax risk : Proof of Theorem 8.3.2

Fix $B \in \mathcal{B}$.

First case : $p(p-1)/2 \geq 4D_B$

Let $\alpha = 3/4$, $\beta = 1/3$, and $\epsilon = D_B/2$. Let $S_B^{\text{disc}}(D_B/2, 3/4)$ the discrete space constructed in Corollary 8.7.1, and the following quantity for $r > 0$:

$$F_B(r) = \left\{rS, S \in S_B^{\text{disc}}\left(\frac{D_B}{2}, \frac{3}{4}\right)\right\}.$$

Let $f^* = \phi(0, \mathbf{\Sigma}^*)$ be the true density. Let $\hat{f}$ be the considered estimator. We define $\tilde{f} = \underset{f \in F_B(r)}{\operatorname{argmin}} \left\{ d_H(\hat{f}, f) \right\}$.

First, we have :

$$d_H(f, \tilde{f}) \leq d_H(f, \hat{f}) + d_H(\hat{f}, \tilde{f}) \leq 2 d_H(\hat{f}, f). \tag{8.20}$$

Secondly, we have :

$$\begin{aligned} d_H(\tilde{f}, f)^2 &\geq 1_{f \neq \tilde{f}} \min_{f' \neq f} d_H(f, f')^2 \\ \mathbb{E}(d_H(\tilde{f}, f)^2) &\geq P(f \neq \tilde{f}) \min_{f' \neq f} d_H(f, f')^2. \end{aligned} \tag{8.21}$$

Then, by combining (8.20) and (8.21) we obtain :

$$\max_{f \in F_B(r)} \mathbb{E}(d_H^2(\hat{f}, f)) \geq \frac{1}{4} \max_{f \in F_B(r)} \left[P_f(f \neq \tilde{f}) \min_{f' \neq f} d_H^2(f, f') \right]. \tag{8.22}$$

We need to design a lower bound for :

$$\max_{f \in F_B(r)} P_f(f \neq \tilde{f}).$$

For this purpose, we use the Birgé Lemma (Lemma 8.7.4) :

Lemma 8.7.4 *Let $(P_f)_{f \in \mathcal{F}}$ a probability family, and $(A_f)_{f \in \mathcal{F}}$ some event pairwise disjoints. Let $a_0 = P_0(A_0)$ and $a = \min_{f \in \mathcal{F}} P_f(A_f)$. Then :*

$$\min_{f \in \mathcal{F}} P_f(A_f) \leq \frac{2e}{2e+1} \vee \frac{\max_{f \in \mathcal{F}} KL(P_f, P_0)}{\log(1 + card(\mathcal{F}))}.$$

Then, if use Birgé's Lemma (Lemma 8.7.4) to control $\max_{f \in F_B(r)} P_f(f \neq \tilde{f})$ in (8.22), we obtain :

$$\max_{f \in F_B(r)} E(d_H^2(\hat{f}, f)) \geq \frac{1}{4(2e+1)} \frac{1}{4 + p \log(e_M / e_m)} \frac{1}{2} \frac{\lambda_m p^3}{e_m^2} D_B r^2. \tag{8.23}$$

if the following inequality is satisfied :

$$\max_{f_1, f_2 \in F_B(r)} (nKL(f_1, f_2)) \leq \frac{2e}{2e+1} \log(1 + \operatorname{card} F_B(r)). \tag{8.24}$$

The inequality (8.24) is satisfied if the inequality (8.25) is fulfilled, with :

$$\frac{n}{2} p^3 \frac{\lambda_m}{e_m^2} D_B r^2 \leq \frac{2e}{2e+1} \left(\rho D_B \log \left(\frac{p(p-1)(e_M - e_m)}{D_B^2} \right) + \kappa K (1 - \log(K)) \right). \tag{8.25}$$

Then, we can replace this condition in (8.23) and we obtain :

$$\max_{f\in F_B(r)} \mathbb{E}(d_H^2(\hat{f}_B, f)) \geq C\frac{D_B}{n}\left(1+\log\frac{C_2}{D_B^2}\right).$$

with :

$$C = \frac{2e}{4(2e+1)^2}\frac{1}{4+p\log(e_M/e_m)}\rho,$$

and with $0.233 \leq \rho \leq 0.234$, and $C_2 = p(p-1)(e_M - e_m)$.

Second case : $p(p-1)/2 \leq 4D_B$

We use the Varshamov-Gilbert lemma (see for example Massart (2007), Lemma 4.7) to construct a discretized space for the covariance, and construct $\tilde{F}_B(r)$ as previously. Then, Birgé's Lemma (Lemma 8.7.4) involves :

$$\sup_{f\in\tilde{F}_B(r)} E(d_H^2(\hat{f}, f) \geq \frac{1}{4(2e+1)}\frac{1}{4+p\log(e_M/e_m)}\frac{1}{2}\frac{\lambda_m p^3}{e_m^2}D_B r^2,$$

if

$$\frac{n}{2}p^3\frac{\lambda_m}{e_m^2}D_B r^2 \leq \frac{2e}{2e+1}\rho\frac{D_B}{2}.$$

Then, we obtain the following bound :

$$\sup_{f\in\tilde{F}_B(r)} E(d_H^2(\hat{f}, f) \geq \frac{1}{4(2e+1)}\frac{1}{4+p\log(e_M/e_m)}(D_B r^2 \wedge \frac{2e}{2e+1}\rho\frac{D_B}{n}).$$

Conclusion

As $F_B(r) \subset F_B^{\text{bound}}$, and $\tilde{F}_B(r) \subset F_B^{\text{bound}}$, choosing $r = (1+\log(C_2/D_B^2))^{1/2}$, we get that :

$$\max_{f\in F_B^{\text{bound}}} \mathbb{E}(d_H^2(\hat{f}_B, f)) \geq C\frac{D_B}{n}\left(1+\log\frac{C_2}{D_B^2}\right),$$

with :

$$C = \frac{2e}{4(2e+1)^2}\frac{1}{4+p\log(e_M/e_m)}\rho,$$

and with $0.233 \leq \rho \leq 0.234$, and $C_2 = p(p-1)(e_M - e_m)$.

Conclusion générale et perspectives

Cette thèse regroupe des contributions méthodologiques utiles à l'analyse des données RNA-seq qui sont discrètes, hétérogènes et présentent une disproportion entre le faible nombre de réplicats biologiques ($n \sim 100$) et le grand nombre de gènes ($p \sim 5000$). Nous avons concentré notre attention sur deux analyses différentes : l'inférence de réseau à l'aide de modèle graphique et l'analyse de co-expression à l'aide de modèle de mélange. Dans un premier temps, nous avons proposé une méthode de filtrage des données RNA-seq supprimant les gènes peu exprimés à l'aide d'un seuil calibré sur les données. Cette méthode permet de réduire le nombre de gènes préliminairement à l'analyse différentielle, mais elle peut aussi s'avérer utile pour l'analyse de co-expression ou l'inférence de réseaux.

Pour l'inférence de réseau, nous avons proposé un modèle graphique hiérarchique log-linéaire de Poisson modélisant le caractère discret et l'importante variabilité inter-échantillon des données RNA-seq. Ce modèle est une alternative à l'utilisation d'un modèle graphique gaussien ou d'un modèle graphique log-linéaire de Poisson sur données transformées. Cependant, les développements autour des modèles graphiques de Poisson sont difficiles car il n'existe pas de manière d'écrire la vraisemblance d'un modèle graphique de Poisson sans imposer des contraintes sévères sur les dépendances modélisées. Dans les modèles graphiques de Poisson, l'estimation des paramètres est effectuée localement, par sélection de voisinage ou *neighborhood selection* en anglais, et les estimateurs ne correspondent pas au maximum de vraisemblance. Des développements théoriques et méthodologiques sont nécessaires pour pouvoir travailler sur un modèle graphique de Poisson joint proprement défini et n'ayant pas les contraintes actuelles du modèle actuellement proposé.

Dans le cadre de l'analyse de co-expression, nous avons proposé de mettre en concurrence deux modélisations des données RNA-seq (poissonienne ou gaussiennes sur données transformées) en les comparant à l'aide d'un critère de sélection de modèle mesurant l'adéquation de ces modèles aux données, calculés à partir des estimateurs du maximum de vraisemblance des modèles respectifs. Des études préliminaires réalisées sur plusieurs jeux de données réelles montrent que le modèle de mélange gaussien semble bien adapté aux données RNA-seq mais des comparaisons sur d'autres jeux de données sont nécessaires pour valider cette conclusion. Plus généralement, la question du choix de transformation est souvent considérée comme une étape de pré-traitement des données alors qu'elle peut s'interpréter comme un problème de choix de modèle. Nous pourrions ainsi déterminer la meilleure transformation possible des données RNA-seq parmi une liste de transformations candidates : transformations Box-Cox, transformations de puissance ou arc sinus hyberbolique. Ce type de comparaison est utile pour l'utilisation du modèle de mélange de lois gaussiennes en analyse de co-expression ou l'utilisation du

modèle graphique gaussien en inférence de réseau des données RNA-seq, mais également pour tout type de données et tout type de modèles probabilistes dont les paramètres sont calculés par maximum de vraisemblance.

Peu importe le type de modélisation, gaussienne ou poissonienne, la performance des méthodes d'inférence de réseaux est limitée par le faible nombre de réplicats à disposition. En pratique, les utilisateurs de ces méthodes doivent sélectionner un sous-ensemble de gènes pour réduire la dimension du problème d'inférence, mais le choix de ces gènes est souvent arbitraire. Dans le cadre du modèle graphique gaussien, nous avons proposé une méthode pour décomposer le problème d'inférence en plusieurs problèmes de taille plus petite à l'aide d'un argument de sélection de modèle non asymptotique. Cette méthode permet de travailler sur un nombre de gènes de départ plus élevé et réduit la part d'arbitraire dans la sélection des variables à inclure dans le réseau. Afin de réduire la dimension du problème, nous aurions également pu adopter le point de vue inverse : plutôt que de décomposer le problème d'inférence en plusieurs groupes et d'inférer le réseau dans chaque groupe (inférence intra-groupe), nous aurions pu agréger les variables entre elles et inférer un réseau de dépendance entre ces groupes (inférence inter-groupe). Il est cependant difficile de définir ces groupes et de les interpréter (modules de gènes co-exprimés ou co-régulés), de définir un critère d'agrégation adéquate pour représenter ces groupes (gènes moyens ou gènes médians), ni d'interpréter le réseau entre ces groupes (régulation d'un module de gènes sur un autre). Dans tous les cas, nous devons rappeler que la dépendance existant entre deux gènes peut prendre de multiples formes et ne peut être résumée à un simple indice statistique.

La méthode de réduction de dimension proposée permet d'inférer des réseaux sur des sous-ensembles de gènes indépendants à partir d'un ensemble de gènes plus large. Cependant, avec un faible nombre de réplicats (e.g. inférieur à une centaine), il est toujours nécessaire d'effectuer une pré-sélection, ce qui nous empêche de travailler sur la totalité des gènes du jeu de données. Afin de pouvoir travailler sur la totalité du jeu de données, sans omettre certains gènes exprimés de l'analyse, nous nous sommes intéressés aux méthodes de classification qui ne souffrent pas du manque de données puisque le problème statistique est inversé par rapport au problème d'inférence de réseaux : les unités statistiques, nombreuses, correspondent aux gènes et les variables, peu nombreuses, correspondent aux réplicats biologiques. Les hypothèses de départ et les objectifs des deux analyses sont différents, mais il serait intéressant, d'un point de vue biologique, de développer des méthodes de détection de gènes co-régulés à partir de modules de gènes co-exprimés.

Afin d'améliorer l'interprétabilité des analyses de co-expression des données RNA-seq par modèle de mélange, nous avons pensé à inclure des variables externes extraites des bases de données d'annotation fonctionnelle de gène dans l'analyse. Nous avons proposé le critère ICAL permettant de sélectionner la partition des données la plus pertinente au regard des données et des annotations externes. Ce critère facilite ainsi l'interprétation de l'analyse de co-expression. Il contribue à améliorer la boucle de rétroaction suivante : les analyses de co-expression successives viennent enrichir les bases de données d'annotation fonctionnelle, et les informations externes de ces bases de données sont à leur tour utilisées pour améliorer la pertinence des nouvelles analyses de co-expression. Cependant, pour que cette boucle de rétroaction soit efficace, le choix des annotations externes à inclure dans l'étape de sélection de modèle doit être effectué par un expert. De manière générale, toute analyse statistique pertinente ne peut être effectuée de manière

isolée, sans une connaissance approfondie des aspects biologiques et bio-informatiques du problème.

Plus généralement, nous avons concentré notre attention sur les données RNA-seq et les bases d'annotation de gènes mais d'autres sources d'information pourraient être incluses. Les sources de données se multiplient et se diversifient : l'intégration de données hétérogènes est un défi important. Par ailleurs, le rôle des données n'est plus de valider une hypothèse déjà formulée en amont. Au contraire, les hypothèses sont formulées à partir des données. La part des décisions arbitraires laissées à l'utilisateur des méthodes (choix de modélisation, choix de modèle, sélection de variables pertinentes) doit être réduite le plus possible afin de ne pas conduire à la formulation de fausses hypothèses et de fausses découvertes. Dans ce contexte, les modèles probabilistes et les outils de sélection de modèle associés offrent un cadre rigoureux d'analyse de données qu'il convient d'exploiter dans les années à venir.

Bibliographie

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6) :716–723.

Allen, G. and Liu, Z. (2012). A log-linear graphical model for inferring genetic networks from high-throughput sequencing data. In *Bioinformatics and Biomedicine (BIBM), 2012 IEEE International Conference on*, pages 1–6.

Allen, G. I. and Liu, Z. (2013). A local Poisson graphical model for inferring networks from sequencing data. *IEEE Transactions on NanoBioscience*, 12(3) :189–198.

Allen, J. D., Xie, Y., Chen, M., Girard, L., and Xiao, G. (2012). Comparing statistical methods for constructing large scale gene networks. *PloS one*, 7(1) :e29348.

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, 11(R106) :1–28.

Anders, S., Pyl, P. T., and Huber, W. (2014). HTSeq A Python framework to work with high-throughput sequencing data. *Bioinformatics*, 31(2) :166–169.

Arlot, S. and Massart, P. (2009). Data-driven calibration of penalties for least-squares regression. *Journal of Machine Learning Research*, 10 :245–279.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology : tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics*, 25(1) :25–9.

Auer, P. and Doerge, R. (2011). A two-stage Poisson model for testing RNA-seq data. *Statistical Applications in Genetics and Molecular Biology*, 10(26) :1–26.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9 :485–516.

Baraud, Y., Giraud, C., and Huet, S. (2009). Gaussian model selection with an unknown variance. *The Annals of Statistics*, 37(2) :630–672.

Basso, K., Margolin, A. a., Stolovitzky, G., Klein, U., Dalla-Favera, R., and Califano, A. (2005). Reverse engineering of regulatory networks in human B cells. *Nature genetics*, 37(4) :382–390.

Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi : An open source software for exploring and manipulating networks. *International AAAI Conference on Weblogs and Social Media*.

Baudry, J.-P., Cardoso, M., Celeux, G., Amorim, M. J., and Ferreira, A. S. (2014). Enhancing the selection of a model-based clustering with external categorical variables. *Advances in Data Analysis and Classification*, 1(1) :1–20.

Baudry, J.-P., Maugis, C., and Michel, B. (2012). Slope heuristics : overview and implementation. *Statistics and Computing*, 22(2) :455–470.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate : a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57 :289–300.

Berend, D. and Tassa, T. (2010). Improved bounds on bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30(2) :185–205.

Bickel, P. J. and Levina, E. (2008). Regularized estimation of large covariance matrices. *The Annals of Statistics*, 36(1) :199–227.

Bien, J. and Tibshirani, R. J. (2011). Sparse estimation of a covariance matrix. *Biometrika*, 98(4) :807–820.

Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7) :719–725.

Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the EM Algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3-4) :561–575.

Biernacki, C., Celeux, G., Govaert, G., and Langrognet, F. (2006). Model-based cluster analysis and discriminant analysis with the MIXMOD software. *Computational Statistics & Data Analysis*, 51 :587–600.

Birgé, L. and Massart, P. (2007). Minimal penalties for Gaussian model selection. *Probability Theory & Related Fields*, 138(1-2).

Birgé, L. (2005). A new lower bound for multiple hypothesis testing. *Information Theory, IEEE Transactions*, 51(4) :1611–1615.

Birney, E., Andrews, T., Bevan, P., Caccamo, M., Chen, Y., Clarke, L., Coates, G., Cuff, J., Curwen, V., Cutts, T., Down, T., Eyras, E., Fernandez-Suarez, X., Gane, P., Gibbins, B., Gilbert, J., Hammond, M., Hotz, H.-R., Iyer, V., Jekosch, K., Kahari, A., Kasprzyk, A., Keefe, D., Keenan, S., Lehvaslaiho, H., McVicker, G.and Melsopp, C., Meidl, P., Mongin, E., Pettett, R., Potter, S., Proctor, G., Rae, M., Searle, S., Slater, G., Smedley, D., Smith, J., Spooner, W., Stabenau, A., Stalker, J., Storey, R., Ureta-Vidal, A., Woodwark, K., Cameron, G., Durbin, R., Cox, A., Hubbard, T., and Clamp, M. (2004). An Overview of Ensembl. *Genome Research*, 14(5) :925–928.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Bottomly, D., Walter, N., Hunter, J., Darakjian, P., Kawane, S., Buck, K., Searles, R. P., Mooney, M., McWeeney, S., and Hitzemann, R. (2011). Evaluating gene expression in C57BL/GJ and DBA/2J mouse striatum using RNA-seq and microarrays. *PLoS One*, 6(3) :e17820.

Bourgon, R., Gentleman, R., and Huber, W. (2010). Independent filtering increases detection power for high-throughput experiments. *PNAS*, 107(21) :9546–9551.

Bouveyron, C., Côme, E., and Jacques, J. (2015). The discriminative functional mixture model for a comparative analysis of bike sharing systems. *The Annals of Applied Statistics*, in press.

Box, G. E. P. and Cox, D. (1964). An analysis of transformations. *Journal of the Royal Statistical Society*, Series B 26(2) :211–252.

Busby, M., Stewart, C., Miller, C. a., Grzeda, K. R., and Marth, G. T. (2013). Scotty : a web tool for designing RNA-Seq experiments to measure differential gene expression. *Bioinformatics*, 29(5) :656–7.

Butte, J., Tamayo, P., Slonim, D., Golub, T. R., and Kohane, I. S. (2000). Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(22) :12182–12186.

Cai, T. T., Zhang, C.-H., and Zhou, H. H. (2010). Optimal rates of convergence for covariance matrix estimation. *The Annals of Statistics*, 38(4) :2118–2144.

Cai, Y., Fendler, B., Atwal, G. S., Biology, Q., Harbor, C. S., and Brook, S. (2012). Utilizing RNA-Seq Data for Cancer Network Inference. In *IEEE International Workshop on Genomic Signal Processing and Statistics*, pages 1–4.

Cánovas, A., Rincon, G., Islas-Trejo, A., Wickramasinghe, S., and Medrano, J. (2010). SNP discovery in the bovine milk transcriptome using RNA-seq technology. *Mammalian Genome*, 21 :592–598.

Castellan, G. (2010). *Sélection d'histogrammes ou de modèles exponentiels de polynômes par morceaux à l'aide d'un critère de type Akaike.* PhD thesis, Université Paris Sud, Orsay.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern recognition*, 28(5) :781–793.

Chaudhuri, S., Drton, M., and Richardson, T. S. (2007). Estimation of a covariance matrix with zeros. *Biometrika*, 94(1) :199–216.

Chen, X., Chen, M., and Ning, K. (2006). BNArray : An R package for constructing gene regulatory networks from microarray data by using Bayesian network. *Bioinformatics*, 22(23) :2952–2954.

Cleveland, W. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368) :829–836.

Dalmay, T. and Edwards, D. R. (2006). MicroRNAs and the hallmarks of cancer. *Oncogene*, 25(46) :6170–5.

Datta, S. (2003). Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, 19(4) :459–466.

De la Fuente, A., Bing, N., Hoeschele, I., and Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18) :3565–3574.

Dempster, A. (1972). Covariance Selection. *Biometrics*, 28(1) :157–175.

Dempster, A., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 39(1) :1–38.

Dillies, M. A., Rau, A., Aubert, J., Hennequet-Antier, C., Jeanmougin, M., Servant, N., Keime, C., Marot, N. S., Castel, D., Estelle, J., Guernec, G., Jagla, B., Jouneau, L., Laloë, D., Le Gall, C., Schaëffer, B., Le Crom, S., Guedj, M., and Jaffrézic, F. (2013). A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics*, 14(6) :671–683.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25) :14863–8.

Erdos, P. and Rényi., A. (1959). On Random Graphs. *Publicationes Mathematicae.*, 6 :419–427.

Frazee, A. C., Langmead, B., and Leek, J. T. (2011). ReCount : a multi-experiment resource of analysis-ready RNA-seq gene count datasets. *BMC Bioinformatics*, 12(449).

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3) :432–441.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1) :1–22.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian Networks to Analyze Expression Data. *Journal of Computational Biology*, 7(3-4) :601–620.

Gallopin, M., Celeux, G., Jaffrézic, F., and Rau, A. (2015). A model selection criterion for model-based clustering of annotated gene expression data. *Statistical Applications in Genetics and Molecular Biology*, 14(5) :413–428.

Gallopin, M., Rau, A., and Jaffrézic, F. (2013). A Hierarchical Poisson Log-Normal Model for Network Inference from RNA Sequencing Data. *PLoS ONE*, 8(10).

Genovese, C. and Wasserman, L. (2000). Rates of convergence for the Gaussian mixture sieve. *The Annals of Statistics*, 28(4) :1105–1127.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J. Y. H., and Zhang, J. (2004). Bioconductor : Open software development for comptuational biology and bioinformatics. *Genome Biology*, 5(R80).

Giorgi, F. M., Del Fabbro, C., and Licausi, F. (2013). Comparative study of RNA-seq and Microarray-derived coexpression networks in *Arabidopsis thaliana*. *Bioinformatics*, 29(6) :717–24.

Giraud, C. (2008). Estimation of Gaussian graphs by model selection. *Electronic Journal of Statistics*, 2 :542–563.

Giraud, C., Huet, S., and Verzelen, N. (2012). Graph selection with GGMselect. *Statistical Applications in Genetics and Molecular Biology*, 11(3).

Graveley, B. R., Brooks, A. N., Carlson, J. W., Duff, M. O., Landolin, J. M., Yang, L., Artieri, C. G., van Baren, M. J., Boley, N., Booth, B. W., Brown, J. B., Cherbas, L., Davis, C. A., Dobin, A., Li, R., Lin, W., Malone, J. H., Mattiuzzo, N. R., Miller, D., Sturgill, D., Tuch, B. B., Zaleski, C., Zhang, D., Blanchette, M., Dudoit, S., Eads, B., Green, R. E., Hammonds, A., Jiang, L., Kapranov, P., Langton, L., Perrimon, N., Sandler, J. E., Wan, K. H., Willingham, A., Zhang, Y., Zou, Y., Andrews, J., Bickel, P. J., Brenner, S. E., Brent, M. R., Cherbas, P., Gingeras, T. R., Hoskins, R. A., Kaufman, T. C., Oliver, B., and Celniker, S. E. (2011). The developmental transcriptome of Drosophila melanogaster. *Nature*, 471(7339) :473–479.

Gregory, P. A., Bert, A. G., Paterson, E. L., Barry, S. C., Tsykin, A., Farshid, G., Vadas, M. A., Khew-Goodall, Y., and Goodall, G. J. (2008). The miR-200 family and miR-205 regulate epithelial to mesenchymal transition by targeting ZEB1 and SIP1. *Nature Cell Biology*, 10(5) :593–601.

Guo, J., Levina, E., Michailidis, G., and Zhu, J. (2011). Joint estimation of multiple graphical models. *Biometrika*, 98(1) :1–15.

Hansen, K., Irizarry, R. A., and Wu, Z. (2012). Removing technical variability in RNA-seq data using conditional quantile normalization. *Biostatistics*, 3 :204–216.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Hastie, T., Tibshirani, R., and Wainwright, M. J. (2015). *Statistical Learning with Sparsity : The Lasso and Generalizations*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability.

Haughton, D. (1988). On the choice of a model to fit data from an exponential family. *The Annals of Statistics*, 16 :342–355.

Hochberg, Y. (1988). A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4) :800–802.

Holmes, I., Harris, K., and Quince, C. (2012). Dirichlet multinomial mixtures : generative models for microbial metagenomics. *PloS ONE*, 7(2) :e30126.

Hong, S., Chen, X., Jin, L., and Xiong, M. (2013). Canonical correlation analysis for RNA-seq co-expression networks. *Nucleic acids research*, 41(8) :e95.

Huang, D., Wei, P., and Pan, W. (2006). Combining gene annotations and gene expression data in model-based clustering : weighted method. *Omics : a journal of integrative biology*, 10(1) :28–39.

Huang, H., Cai, L., and Wong, W. H. (2008). Clustering analysis of SAGE transcription profiles using a Poisson approach. *Methods in molecular biology*, 387(7) :185–198.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1) :193–218.

Iancu, O. D., Kawane, S., Bottomly, D., Searles, R., Hitzemann, R., and McWeeney, S. (2012). Utilizing RNA-Seq data for de novo coexpression network inference. *Bioinformatics*, 28(12) :1592–7.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37 :547–549.

Jiang, D., Tang, C., and Zhang, A. (2004). Cluster analysis for gene expression data : a survey. *IEEE Transactions on Knowledge and Data Engineering*, 16(11) :1370–1386.

Kanehisa, M. and Goto, S. (2000). KEGG : Kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1) :27–30.

Karlis, D. (2003). An EM algorithm for multivariate Poisson distribution and related models. *Journal of Applied Statistics*, 30(1) :63–77.

Karlis, D. and Meligkotsidou, L. (2005). Multivariate Poisson regression with covariance structure. *Statistics and Computing*, 15(4) :255–265.

Kasprzyk, A.and Keefe, D., Smedley, D., London, D., Spooner, W., Melsopp, C., Hammond, M., Rocca-Serra, P., Cox, T., and Birney, E. (2004). EnsMart : A generic system for fast and flexible access to biological data. *Genome Research*, 14(1) :160–169.

Kovalchuk, O., Filkowski, J., Meservy, J., Ilnytskyy, Y., Tryndyak, V. P., Chekhun, V. F., and Pogribny, I. P. (2008). Involvement of microRNA-451 in resistance of the MCF-7 breast cancer cells to chemotherapeutic drug doxorubicin. *Molecular Cancer Therapeutics.*, 7(7) :2152–9.

Łabaj, P., Leparc, G., Linggi, B., Markillie, L., Wiley, H., and Kreil, D. (2011). RNA-seq precision in quantitative expression profiling. *Bioinformatics*, 27(13) :i383–i381.

Langfelder, P. and Horvath, S. (2008). WGCNA : an R package for weighted correlation network analysis. *BMC Bioinformatics*, 9 :559.

Lauritzen, S. L. (1996). *Graphical Models*. Clarendon Press.

Law, C. W., Chen, Y., Shi, W., and Smyth, G. K. (2014). Voom : precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome biology*, 15(2) :R29.

Le, Y. and Hastie, T. (2014). Sparse Quadratic Discriminant Analysis and Community Bayes. *arXiv preprint arXiv : :1407.4543*.

Lebarbier, E. (2005). Detecting multiple change-points in the mean of gaussian process by model selection. *Signal Processing*, 85(4) :717 – 736.

Lebarbier, E. and Mary-Huard, T. (2006). Le critère BIC : fondements théoriques et interprétation. *Journal de la SFdS*, 147(1) :39–57.

Lebret, R., Iovleff, S., Langrognet, F., Biernacki, C., Celeux, G., and Govaert, G. (2013). Rmixmod : The R Package of the Model-Based Unsupervised, Supervised and Semi-Supervised Classification Mixmod Library. *Journal of Statistical Software*, In revision.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16) :2078–2079.

Li, J., Witten, D. M., Johnstone, I. M., and Tibshirani, R. (2012). Normalization, testing, and false discovery rate estimation for RNA sequencing data. *Biostatistics*, 13(3) :523–38.

Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdóttir, H., Tamayo, P., and Mesirov, J. P. (2011). Molecular signatures database (MSigDB) 3.0. *Bioinformatics*, 27(12) :1739–40.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. *Genome biology*, 15(550) :1–21.

Mach, N., Berri, M., Esquerré, D., Chevaleyre, C., Lemonnier, G., Billon, Y., Lepage, P., Oswald, I. P., Doré, J., Rogel-Gaillard, C., and Estellé, J. (2014). Extensive expression differences along porcine small intestine evidenced by transcriptome sequencing. *PloS ONE*, 9(2) :e88515.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *5th Berkeley Symposium on Mathematical Statistics and Probability 1967*, 1(233) :281–297.

Margolin, A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla Favera, R., and Califano, A. (2006). ARACNE : an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7 Suppl 1 :S7.

Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M., and Gilad, Y. (2008). comparison with gene expression arrays RNA-seq : An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, 18(9) :1509–1517.

Massart, P. (2007). *Concentration inequalities and model selection.* Lecture Notes in Mathematics. Springer, 33, 2003, Saint-Flour, Cantal.

Maugis, C. and Michel, B. (2011). A non asymptotic penalized criterion for Gaussian mixture model selection. *ESAIM. Probability and Statistics.*, 15 :41–68.

Mazumder, R. and Hastie, T. (2012). Exact covariance thresholding into connected components for large-scale Graphical Lasso. *Journal of Machine Learning Research*, 13 :781–794.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models, Willey Series in Probability and Statistics*. John Wiley & Sons, New York.

Meinshausen, N. and Buhlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3) :1436–1462.

Meyer, P. E., Lafitte, F., and Bontempi, G. (2008). minet : A R/Bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics*, 9 :461.

Morlini, I. (2011). A latent variables approach for clustering mixed binary and continuous variables within a Gaussian mixture model. *Advances in Data Analysis and Classification*, 6(1) :5–28.

Mortazavi, A., Williams, B., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5(7) :621–628.

Novère, N. L. (2015). Quantitative and logic modelling of molecular and gene networks. *Nature Publishing Group*, 16(February) :146–158.

Oshlack, A., Robinson, M., and Young, M. D. (2010). From RNA-seq reads to differential expression results. *Genome Biology*, 11(220).

Oshlack, A. and Wakefield, M. J. (2009). Transcript length bias in RNA-seq data confounds systems biology. *Biology Direct*, 4(14).

Pan, W. (2006). Incorporating gene functions as priors in model-based clustering of microarray gene expression data. *Bioinformatics*, 22(7) :795–801.

Pearl, J. (1990). *Causality*. Cambridge University Press.

Peng, J., Wang, P., Zhoo, N., and Zhu, J. (2009). Partial Correlation Estimation by Joint Sparse Regression Models. *Journal of American Statistical Associations*, 104(486) :735–746.

Peter, M. E. (2009). Let-7 and miR-200 microRNAs : Guardians against pluripotency and cancer progression. *Cell Cycle.*, 8(6) :843–52.

Pickrell, J. (2010). Understanding mechanisms underlying human gene expression variation with RNA sequencing. *Nature*, 464(7289) :768–772.

R Development Core Team (2009). *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Ramsköld, D., Wang, E. T., Burge, C. B., and Sandberg, R. (2009). An abundance of ubiquitously expressed genes revealed by tissue transcriptome sequence data. *PLoS Computational Biology*, 5(12) :e1000598.

Rau, A. (2010). *Reverse engineering gene networks using genomic time-course data*. PhD thesis, Purdue University.

Rau, A., Gallopin, M., Celeux, G., and Jaffrézic, F. (2013). Data-based filtering for replicated high-throughput transcriptome sequencing experiments. *Bioinformatics*, 29(17) :2146–2152.

Rau, A., Maugis-Rabusseau, C., Martin-Magniette, M.-L., and Celeux, G. (2015). Co-expression analysis of high-throughput transcriptome sequencing data with Poisson mixture models. *Bioinformatics*, 31(9) :1420–1427.

Reverter, A. and Chan, E. K. F. (2008). Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks. *Bioinformatics*, 24(21) :2491–2497.

Risso, D., Schwartz, K., Sherlock, G., and Dudoit, S. (2011). GC-content normalization for RNA-seq data. *BMC Bioinformatics*, 12(480).

Robinson, M. and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(R25).

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edgeR : a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1) :139–40.

Sam, L., Lipson, D., Raz, T., Cao, X., Thompson, J., Milos, P., Robinson, D., Chinnaiyan, A., Kumar-Sinha, C., and Maher, C. (2011). A comparison of single molecule and amplification based sequencing of cancer transcriptomes. *PLoS One*, 6(3) :e17305.

Sanger, F., Nicklen, S., and Coulson, a. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12) :5463–5467.

Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4 :Article32.

Schelldorfer, J., Meier, L., and Bühlmann, P. (2014). Glmmlasso : An algorithm for high-dimensional generalized linear mixed models using 1-penalization. *Journal of Computational and Graphical Statistics*, 23(2) :460–477.

Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235) :467–70.

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2) :461–464.

Severin, A. J., Woody, J. L., Bolon, Y.-T., Joseph, B., Diers, B. W., Farmer, A. D., Muehlbauer, G. J., Nelson, R. T., Grant, D., Specht, J. E., Graham, M. a., Cannon, S. B., May, G. D., Vance, C. P., and Shoemaker, R. C. (2010). RNA-Seq Atlas of Glycine max : a guide to the soybean transcriptome. *BMC plant biology*, 10(2007) :160.

Shapiro S. S. and Wilk M. B. (1965). An analysis of variance test for normality. *Biometrika*, 52(3 and 4) :561.

Si, Y., Liu, P., Li, P., and Brutnell, T. P. (2013). Model-based clustering for RNA-seq data. *Bioinformatics*, 30(2) :197–205.

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3.

Srivastava, N., Manvati, S., Srivastava, A., Pal, R., Kalaiarasan, P., Chattopadhyay, S., Gochhait, S., Dua, R., and Bamezai, R. N. K. (2011). miR-24-2 controls H2AFX expression regardless of gene copy number alteration and induces apoptosis by targeting antiapoptotic gene BCL-2 : a potential for therapeutic intervention. *Breast Cancer Research.*, 13(2) :R39.

Stahlhut Espinosa, C. E. and Slack, F. J. (2006). The role of microRNAs in cancer. *Yale Journal of Biology and Medicine*, 79(3-4) :131–40.

Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., Iyer, R., Schatz, M. C., Sinha, S., and Robinson, G. E. (2015). Big Data : Astronomical or Genomical ? *PLOS Biology*, 13(7) :e1002195.

Steuer, R., Humburg, P., and Selbig, J. (2006). Validation and functional annotation of expression-based clusters based on gene ontology. *BMC Bioinformatics*, 7 :380.

Strub, T., Giuliano, S., Ye, T., Bonet, C., Keime, C., Kobi, D., Le Gras, S., Cormont, M., Ballotti, R., Bertolotto, C., and Davidson, I. (2011). Essential role of microphthalmia transcription factor for DNA replication, mitosis and genomic stability in melanoma. *Oncogene*, 30 :2319–2332.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis : a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43) :15545–50.

Sultan, M., Schulz, M. H., Richard, H., Magen, A., Klingenhoff, A., Scherf, M., Seifert, M., Borodina, T., Soldatov, A., Parkhomchuk, D., Schmidt, D., Keeffe, S. O., Haas, S., Vingron, M., Lehrach, H., and Yaspo, M.-l. (2008). of the Human Transcriptome. *Science*, 685(August) :956–960.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., and Golub, T. R. (1999). Interpreting patterns of gene expression with self-organizing maps : methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences of the United States of America*, 96(6) :2907–2912.

Tan, K., Witten, D., and Shojaie, A. (2015). The Cluster Graphical Lasso for improved estimation of Gaussian graphical models. *Computational Statistics & Data Analysis*, 85 :23–36.

Tari, L., Baral, C., and Kim, S. (2009). Fuzzy c-means clustering with prior biological knowledge. *Journal of Biomedical Informatics*, 42(1) :74–81.

Thomas, I., Frankhauser, P., and Biernacki, C. (2008). The Fractal Morphology of the Built-Up Landscape. *Landscape of Urban Plan,*, 84(2) :99–115.

Tipney, H. and Hunter, L. (2010). An introduction to effective use of enrichment analysis software. *Human Genomics*, 4(3) :202.

Trapnell, C., Pachter, L., and Salzberg, S. L. (2009). TopHat : Discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9) :1105–1111.

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J., and Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology*, 28(5) :511–5.

Verbanck, M., Lê, S., and Pagès, J. (2013). A new unsupervised gene clustering algorithm based on the integration of biological knowledge into expression data. *BMC Bioinformatics*, 14 :42.

Verzelen, N. (2012). Minimax risks for sparse regressions : Ultra-high dimensional phenomenons. *Electronic Journal of Statistics*, 6 :38–90.

Wang, L., Feng, Z., Wang, X., and Zhang, X. (2010). DEGseq : an R package for idenetifying differentially expressed genes from RNA-seq data. *Bioinformatics*, 26 :136–138.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301) :236–244.

Werhli, A. V., Grzegorczyk, M., and Husmeier, D. (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics*, 22(20) :2523–2531.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics*. Wiley Publishing.

Wickham, H. (2009). *ggplot2 : elegant graphics for data analysis*. Springer, New York.

Wille, A. and Peter, B. (2006). Low-order conditional independence graphs for inferring genetic networks low-order conditional independence graphs for inferring genetic networks. *Statistical Applications in Genetics and Molecular Biology*, 5(1).

Witten, D. M. (2011). Classification and clustering of sequencing data using a poisson model. *The Annals of Applied Statistics*, 5(4) :2493–2518.

Witten, D. M., Friedman, J. H., and Simon, N. (2011). New insights and faster computations for the Graphical Lasso. *Journal of Computational and Graphical Statistics*, 20(4) :892–900.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, a. E., and Ruzzo, W. L. (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10) :977–87.

Zou, C. and Xu, Q. (2012). miR-145 inhibits tumor angiogenesis and growth by N-RAS and VEGF. *Cell Cycle*, 11(11) :2137–45.

Titre : Classification et inférence de réseau pour les données RNA-seq.

Mots clés : modèle de mélange, modèle graphique, données RNA-seq, classification, inférence de réseaux, sélection de modèle

Résumé : Cette thèse regroupe des contributions méthodologiques à l'analyse statistique des données issues des technologies de séquençage du transcriptome (RNA-seq). Les difficultés de modélisation des données de comptage RNA-seq sont liées à leur caractère discret et au faible nombre d'échantillons disponibles, limité par le coût financier du séquençage. Une première partie de travaux de cette thèse porte sur la classification à l'aide de modèles de mélange. L'objectif de la classification est la détection de modules de gènes co-exprimés. Un choix naturel de modélisation des données RNA-seq est un modèle de mélange de lois de Poisson. Mais des transformations simples des données permettent de se ramener à un modèle de mélange de lois gaussiennes. Nous proposons de comparer, pour chaque jeu de données RNA-seq, les différentes modélisations à l'aide d'un critère objectif permettant de sélectionner la modélisation la plus adaptée aux données. Par ailleurs, nous présentons un critère de sélection de modèle prenant en compte des informations biologiques externes sur les gènes. Ce critère facilite l'obtention de classes biologiquement interprétables. Il n'est pas spécifique aux données RNA-seq. Il est utile à toute analyse de co-expression à l'aide de modèles de mélange visant à enrichir les bases de données d'annotations fonctionnelles des gènes. Une seconde partie de travaux de cette thèse porte sur l'inférence de réseau à l'aide d'un modèle graphique. L'objectif de l'inférence de réseau est la détection des relations de dépendance entre les niveaux d'expression des gènes. Nous proposons un modèle d'inférence de réseau basé sur des lois de Poisson, prenant en compte le caractère discret et la grande variabilité inter-échantillons des données RNA-seq. Cependant, les méthodes d'inférence de réseau nécessitent un nombre d'échantillons élevé. Dans le cadre du modèle graphique gaussien, modèle concurrent au précédent, nous présentons une approche non-asymptotique pour sélectionner des sous-ensembles de gènes pertinents, en décomposant la matrice variance en blocs diagonaux. Cette méthode n'est pas spécifique aux données RNA-seq et permet de réduire la dimension de tout problème d'inférence de réseau basé sur le modèle graphique gaussien.

Title : Clustering and network inference for RNA-seq data.

Keywords : mixture model, graphical model, RNA-seq data, clustering, network inference, model selection

Abstract : This thesis gathers methodologicals contributions to the statistical analysis of next-generation high-throughput transcriptome sequencing data (RNA-seq). RNA-seq data are discrete and the number of samples sequenced is usually small due to the cost of the technology. These two points are the main statistical challenges for modelling RNA-seq data. The first part of the thesis is dedicated to the co-expression analysis of RNA-seq data using model-based clustering. A natural model for discrete RNA-seq data is a Poisson mixture model. However, a Gaussian mixture model in conjunction with a simple transformation applied to the data is a reasonable alternative. We propose to compare the two alternatives using a data-driven criterion to select the model that best fits each dataset. In addition, we present a model selection criterion to take into account external gene annotations. This model selection criterion is not specific to RNA-seq data. It is useful in any co-expression analysis using model-based clustering designed to enrich functional annotation databases. The second part of the thesis is dedicated to network inference using graphical models. The aim of network inference is to detect relationships among genes based on their expression. We propose a network inference model based on a Poisson distribution taking into account the discrete nature and high inter sample variability of RNA-seq data. However, network inference methods require a large number of samples. For Gaussian graphical models, we propose a non-asymptotic approach to detect relevant subsets of genes based on a block-diagonale decomposition of the covariance matrix. This method is not specific to RNA-seq data and reduces the dimension of any network inference problem based on the Gaussian graphical model.