Reconstruction and clustering with graph optimization and priors on gene networks and images

Aurélie Pirayre

École Doctorale Mathématiques et Sciences et Technologies de l'Information et de la Communication

Université Paris-Est

THÈSES

Spécialité: Traitement du Signal et Image

présentée par Aurélie PIRAYRE

soutenue le 03 juillet 2017

Reconstruction et Classification avec Optimisation sur graphes et a priori pour Réseaux de gènes et Images

Rapporteurs:	Pascal FROSSARD	*EPFL*
	Jean-Philippe VERT	*Mines ParisTech*
Examinateurs:	Stéphane ROBIN	*INRA*
	Hugues TALBOT	*Labinfo IGM*
Directeur de thèse:	Jean-Christophe PESQUET	*CentraleSupélec*
Co-encadrants:	Laurent DUVAL	*IFP Energies nouvelles*
	Camille COUPRIE	*Facebook A.I. Research*
	Frédérique BIDARD-MICHELOT	*IFP Energies nouvelles*

"J'entends, j'oublie.

Je vois, je me souviens.

Je fais, je comprends"

Attribué à Confucius

Abstract

The discovery of novel gene regulatory processes improves the understanding of cell phenotypic responses to external stimuli for many biological applications, such as medicine, environment or biotechnologies. To this purpose, transcriptomic data are generated and analyzed from DNA microarrays or more recently RNAseq experiments. They consist in genetic expression level sequences obtained for all genes of a studied organism placed in different living conditions. From these data, gene regulation mechanisms can be recovered by revealing topological links encoded in graphs. In regulatory graphs, nodes correspond to genes. A link between two nodes is identified if a regulation relationship exists between the two corresponding genes. Such networks are called Gene Regulatory Networks (GRNs). Their construction as well as their analysis remain challenging despite the large number of available inference methods.

In this thesis, we propose to address this network inference problem with recently developed techniques pertaining to graph optimization. Given all the pairwise gene regulation information available, we propose to determine the presence of edges in the final GRN by adopting an energy optimization formulation integrating additional constraints. Either biological (information about gene interactions) or structural (information about node connectivity) *a priori* have been considered to restrict the space of possible solutions. Different priors lead to different properties of the global cost function, for which various optimization strategies, either discrete and continuous, can be applied. The post-processing network refinements we designed led to computational approaches named *BRANE* for "Biologically-Related A priori for Network Enhancement". For each of the proposed methods — *BRANE Cut*, *BRANE Relax* and *BRANE Clust* — our contributions are threefold: a priori-based formulation, design of the optimization strategy and validation (numerical and/or biological) on benchmark datasets from DREAM4 and DREAM5 challenges showing numerical improvement reaching 20 %.

In a ramification of this thesis, we slide from graph inference to more generic data processing such as inverse problems. We notably invest in *HOGMep*, a Bayesian-based approach using a Variational Bayesian Approximation framework for its resolution. This approach allows to jointly perform reconstruction and clustering/segmentation tasks on multi-component data (for instance signals or images). Its performance in a color image deconvolution context demonstrates both quality of reconstruction and segmentation. A preliminary study in a medical data classification context linking genotype and phenotype yields promising results for forthcoming bioinformatics adaptations.

Résumé

Le couplage entre des phénomènes croissants de pollution mondiale, de gaz à effet de serre, de réchauffement climatique et de diminution des ressources énergétiques fossiles soulève des problématiques environnementales pour le futur, nécessitant de ce fait le développement de nouvelles énergies, dites alternatives. C'est le cas des biocarburants, et notamment le bioéthanol, qui connait maintenant un regain d'intérêt.

Alors que les biocarburants de première génération — obtenus à partir de cultures sucrières et amylacées — sont vivement controversés en raison de leur compétitivité avec la filière agroalimentaire, un attachement particulier a été donné au développement des biocarburants dits de seconde génération. Ces derniers sont obtenus à partir de biomasse lignocellulosique (végétaux non comestibles ou résidus). Le procédé classique de production de bioéthanol suivant le procédé de seconde génération consiste en trois grandes étapes : *i*) un pré-traitement permettant d'extraire la cellulose — un polymère de glucoses — contenue dans la biomasse, *ii*) une hydrolyse de la cellulose en monomères de glucose, cette hydrolyse étant réalisée par un cocktail d'enzymes dédiées et enfin *iii*) une fermentation des molécules de glucose en éthanol. Cependant, la production d'enzymes et la phase d'hydrolyse représentent à elles seules quelques 30 % du coût de l'éthanol produit, limitant ainsi la viabilité économique du procédé. Une recherche active est donc nécessaire pour améliorer à moindre coût la production d'enzymes.

La production d'enzymes cellulolytiques nécessaire à la conversion cellulose/sucre se fait, d'après le choix des acteurs industriels, par un champignon filamenteux, *Trichoderma reesei*. Afin d'améliorer ses rendements de production, une optimisation génétique de ce champignon peut être envisagée. C'est notamment ce qui a été fait au cours des années 1980, par l'utilisation de mutagenèse aléatoires. Ces manipulations génétiques ont permis de sélectionner des souches hyper-productrices. Cependant, l'utilisation de mutagenèses aléatoires semble avoir maintenant atteint ses limites et des approches dirigées sont à privilégier. Une optimisation génétique par mutagenèse dirigée requiert cependant d'avoir une bonne connaissance du processus de production d'enzymes par le champignon. L'information, trop parcimonieuse, que nous avons sur les mécanismes fins de *T. reesei* nous amène donc dans un premier temps à mieux connaître et comprendre le fonctionnement génétique de ce champignon lors de sa production d'enzymes cellulolytiques. Les biologistes recourent aux données "-omiques", qui offrent un accès sans précédent à des mécanismes biologiques fondamentaux, à différentes échelles. Les données, générées en volume important, font appel à des compétences pluridisciplinaires, à l'intersection des biotechnologies et du développement d'analyse algorithmique, pour une intégration et une interprétation effectives.

Partant du postulat que la production de protéines (que sont les enzymes) est liée à l'expression des gènes sous-jacents, la compréhension du mécanisme de production de protéines peut être obtenue par celle des mécanismes d'expression des gènes et donc leur régulation. La régulation des gènes fait elle-même intervenir des protéines, issues elles-mêmes de gènes. On comprend alors que la détection d'interactions entre gènes permet de comprendre leurs mécanismes de régulation et donc d'expression menant à terme aux protéines. Pour ce faire, les études transcriptomiques nous permettent d'avoir accès, pour une population de cellules données dans des conditions expérimentales bien choisies, au niveau d'expression de tous les gènes. En recueillant les niveaux d'expression des gènes pour ces différentes conditions expérimentales, des profils d'expression des gènes sont ainsi obtenus. À partir de ces profils d'expression, il est alors possible après traitements d'en déduire des interactions entre gènes. Ces interactions peuvent être modélisées sous la forme de graphes, où les nœuds correspondent aux gènes et les liens entre les nœuds aux interactions entre gènes. De tels graphes sont appelées des Réseaux de Régulation de Gènes (RRGs). C'est dans ce contexte que cette thèse s'inscrit, où les contributions proposées portent sur le développement d'outils bio-informatiques visant à construire des RRGs à partir de données transcriptomiques. Cette partie introductive est notamment détaillée dans le chapitre 2.

La construction de RRGs à partir de données transcriptomiques peut être vue comme un procédé en deux étapes : *i*) calcul d'un poids pour chaque arête du graphe complet et *ii*) seuillage de ces poids pour garder les liens significatifs. Comme le détaille l'étude bibliographique du chapitre 3, le développement de méthodes d'inférence de RRGs porte essentiellement sur l'étape de calcul du poids. Afin de compléter une méthode de calcul de poids satisfaisante, nous avons concentré nos efforts sur le développement de méthodes de sélection d'arêtes, plus puissantes qu'un simple seuillage sur les poids. Pour ce faire, le problème de seuillage classique a été formulé à l'aide d'une fonction objectif à optimiser, qui dépend de variables binaires portant sur chaque arête et témoignant de la présence ou de l'absence de l'arête dans le graphe final. La résolution du problème ainsi formulé peut paraître triviale mais cette formulation donne ainsi une base pour de potentielles améliorations, notamment par l'ajout de termes de régularisation bien choisis : notre démarche a été d'encoder, à travers ces termes de régularisation additionnels, des *a priori* biologiques sur les mécanismes de régulation des gènes et/ou structuraux sur les réseaux attendus. Les différents *a priori* choisis ont donné lieu à des fonctions objectifs dont les propriétés requièrent le choix d'algorithmes dédiés. Les différents *a priori* biologiques que nous avons formulés font état d'une connaissance préalable sur des gènes codant pour des protéines appelées facteurs de transcription. Ces protéines sont des acteurs de premier plan dans la régulation des gènes et l'information qu'elles portent est donc à promouvoir. Ce travail de thèse a mené à un ensemble d'approches computationnelles nommé *BRANE*, pour "Biologically Related A priori for Network Enhancement". Les différentes méthodes de sélection d'arêtes développées dans cette thèse peuvent être perçues comme des méthodes de post-traitement à utiliser sur des graphes pleinement connectés et pondérés.

Le chapitre 4 est dédié à la présentation de *BRANE Cut*, notre première stratégie de sélection d'arêtes. En plus de sélectionner les arêtes fortement pondérées comme dans le seuillage classique, la fonction objectif que nous avons conçue permet de promouvoir une structure modulaire

dans les réseaux inférés. Par ailleurs, un *a priori* de co-régulation de gènes est également pris en compte par l'ajout d'un terme de régularisation permettant un couplage dans l'inférence d'arêtes mettant en jeu des couples de facteurs de transcription agissant en coopération. La formulation finale du problème prend la forme d'une fonction objectif ressortissant aux problèmes de coupe minimale dans un graphe. Par dualité (*minimum cut/maximal flow*), notre problème d'optimisation discrète est résolu grâce à l'algorithme de flot maximal. Les performances de *BRANE Cut* ont été validées sur des données simulées issues des challenges DREAM4 et DREAM5 avant que d'être également validées sur données réelles provenant d'un organisme bactérien tel que *Escherichia coli* ou de notre champignon d'étude *Trichoderma reesei*. En complément d'une validation de la méthode, des comparaisons avec des méthodes état de l'art telles que CLR, GENIE3 ou encore le post-traitement Network Deconvolution (ND) ont permis de mettre en évidence les améliorations fournies par *BRANE Cut*, tant sur le plan de la performance numérique (avec des améliorations atteignant environ 11 %) que de l'interprétation biologique des réseaux inférés.

Dans le même état d'esprit que *BRANE Cut*, une seconde stratégie, nommée *BRANE Relax*, a été développée. Le chapitre 5 lui est consacré. Comme précédemment, la fonction objectif définie favorise la sélection d'arêtes de fort poids en plus de fournir un réseau modulaire. Dans cette approche, l'*a priori* de co-régulation a été remplacé par un *a priori* sur la connectivité des gènes autres que ceux identifiés comme codant pour un facteur de transcription. La formulation résultante, dans sa forme discrète, ne peut être optimisée par des algorithmes d'optimisation combinatoire. En revanche, en relaxant le problème dans le domaine continu, il est alors possible de le résoudre à l'aide d'un algorithme de gradient projeté. Cependant ce type d'algorithme, connu pour sa potentielle lenteur de convergence dans le cas de problèmes de grandes dimensions, peut être accéléré par l'introduction de matrices de pré-conditionnement issues du principe de Majoration-Minimisation couplée à des stratégies par blocs. L'approche proposée a été validée et comparée à des méthodes de l'état de l'art (CLR, GENIE3 et le post-traitement ND) sur des données synthétiques de parangonnage issues des challenges DREAM4 et DREAM5 et montre des améliorations pouvant atteindre 8 %, environ.

En complément de l'inférence de réseaux, la classification des gènes par rapport à leurs profils d'expression est également une pratique très courante dans le traitement de données transcriptomiques. Cette classification a pour but de regrouper les gènes ayant des profils d'expression similaires, au sens d'un certain critère. Ces groupes de gènes sont ensuite étudiés plus en détail afin de déterminer si des fonctions particulières ressortent de ces groupes de gènes, pouvant potentiellement appartenir à une même voie biologique. Cependant, cette classification est souvent menée de façon indépendante à l'inférence de réseaux. Afin d'améliorer l'inférence et son interprétation, l'intégration d'une information de groupement des gènes est proposée dans *BRANE Clust*. En effet, comme détaillé dans le chapitre 6 dédié à *BRANE Clust*, la fonction objectif que nous proposons a été conçue pour pénaliser les arêtes liant des nœuds appartenant à des clusters distincts. Pour ce faire, en complément des variables binaires sur les arêtes, des variables discrètes (mais non nécessairement binaires) sont également définies sur les nœuds. Ces variables encodent le label de la partition auquel le nœud est assigné. Par conséquent, la classification n'est pas calculée de façon indépendante mais est couplée à l'inférence. Une contrainte sur la construction de classes centrées sur les facteurs de transcription permet de favoriser une struc-

ture modulaire dans le réseau final. Une stratégie d'optimisation alternée peut être mise en place pour résoudre ce problème. Le sous-problème portant sur l'inférence à proprement parler peut se résoudre de façon explicite, alors que le sous-problème de classification peut être résolu, après relaxation, par une résolution de systèmes linéaires. Cette approche a été validée à la fois sur des données synthétiques et réelles issues des challenges DREAM4 et DREAM5. Des améliorations par rapport aux méthodes états de l'art (CLR, GENIE3 et ND) ont également été démontrées, autant en termes de performances numériques (avec des gains atteignant 20 %) qu'en termes d'interprétations biologiques faites sur un réseau inféré à partir de données sur la bactérie *Escherichia coli*.

Ce travail de thèse a donc permis le développement de deux méthodes principales (*BRANE Cut* and *BRANE Clust*) et d'une plus intermédiaire (*BRANE Relax*) pour la sélection d'arêtes dans le contexte de réseaux de régulation de gènes. Ces méthodes se basent sur une formulation variationnelle d'un problème d'optimisation intégrant des *a priori* biologiques et/ou structuraux. Ces méthodes, qui peuvent être utilisées en post-traitement des méthodes classiques d'inférence, ont su faire leurs preuves sur des données synthétiques aussi bien que réelles. Cependant, en complément de ce travail essentiellement orienté sur l'inférence de réseaux de régulation de gènes, nous avons mené des travaux vers des traitements de graphes plus génériques, dans le contexte des problèmes inverses. Ce travail préliminaire, présenté dans le chapitre 7, a été pensé en vue d'adaptations à des problématiques plus larges, incluant la biologie. Il a permis de valoriser un travail générique autour d'*HOGMep*, une méthode bayésienne développée pour effectuer conjointement des tâches de restauration et de classification sur des données multi-composantes. Les performances d'*HOGMep* ont été éprouvées et validées dans deux contextes très distincts. Une première application en déconvolution d'images couleur a d'abord été abordée. Des améliorations, tant sur le plan de la reconstruction que celui de la segmentation, ont ainsi pu être démontrées. Enfin, son utilité pour la classification de données d'expression de gènes dans un contexte médical de relations génotype/phénotype a également été établie. La validation de ces performances est une première étape vers une adaptation potentielle de *HOGMep* à des problèmes de biologie plus poussés.

Enfin, un récapitulatif des contributions réalisées durant cette thèse ainsi que plusieurs perspectives sont présentés dans le chapitre 8.

Acronyms

AIC Akaike Information Criterion.

ANOVA ANalysis Of Variance.

ARACNE Algorithm for the Reconstruction of Accurate Cellular NEtwork.

BIC Bayesian Information Criterion.

BN Bayesian network.

BRANE Biologically-Related A priori for Network Enhancement.

C3Net Conservative Causal Core.

CAST Cluster Affinity Search Technique.

cDNA complementary DNA.

CLR Context Likelihood of Relatedness.

CMI Conditional Mutual Information.

CPM Counts Per Million.

DINGO Differential network analysis in genomics.

DNA DesoxyriboNucleic Acid.

DPI Data Processing Inequality.

DREAM Dialogue on Reverse Engineering Assessment and Methods.

EM Expectation-Maximization.

FN False Negative.

FP False Positive.

GEO Gene Expression Omnibus.

GGM Gaussian Graphical Models.

GH Glycosyl Hydrolase.

GNW GeneNetWeaver.

GO Gene Ontology.

GRN Gene Regulatory Network.

KEGG Kyoto Encyclopedia of Genes and Genomes.

lasso Least Absolute Shrinkage and Selection Operator.

LOWESS LOcally WEighted Scatterplot Smoothing.

Med Median.

MEME Multiple EM for Motif Elicitation.

MeV MultiExperiment Viewer.

MI3 Mutual Information 3.

MIC Maximal Information Coefficient.

MM Majorize-Minimize.

MRMR Maximum Relevance/Minimum Redundancy.

mRNA messenger RNA.

MRNET Minimum Redundancy NETworks.

NB negative binomial.

NGS Next Generation Sequencing.

PCR Polymerization Chain Reaction.

PGM Probabilistic Graphical Model.

PMT photomultiplier.

Q Quantile.

RMA Robust Multichip Averaging.

RN Relevance Network.

ROC Receiver Operator Characteristics.

RPKM Reads Per Kilobase per Million mapped reads.

RSAT Regulatory Sequence Analysis Tools.

SAM Significance Analysis of Microarrays.

SCAD Smoothly Clipped Absolute Deviation.

SIMoNe Statistical Inference for Modular Networks.

SNP Single Nucleotide Polymorphisme.

SVD Singular Value Decomposition.

TC Total Counts.

TGD Threshold Gradient Descent.

TMM Trimmed Mean of `M`-values.

TN True Negative.

TP True Positive.

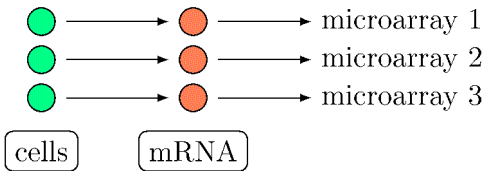
UQ Upper Quartile.

VI Variation of Information.

WGCNA Weighted correlation network analysis.

Glossary

biological replicates for a given experimental condition, different cultures of same cells are prepared in parallel
.



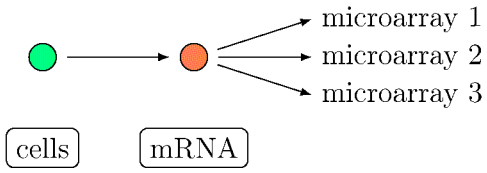
dual knockdown steady-state level for two simultaneously deleted genes.

knockdown steady-state level of a single-gene knockdown leading to a transcription rate arbitrary decreased to twice.

knockout steady-state level of a deleted genes leading to a gene transcription rate equals to 0.

multifactorial steady-state levels of all genes after multifactorial perturbations. This simulation tends to simultaneously increase or decrease all basal expression level by different random amounts.

technical replicates for a given experimental condition, a unique cell culture is firstly processed and split just before hybridization
.



wild type steady-state level of the unperturbed gene.

Contents

1 Introduction

1.1 Context and motivations

The emergence of industrial bio-processes represents a major challenge, in the context of the energy transition or the "Nouvelle France Industrielle (NFI)" project, for instance. Related research activities include production processes for second-generation bio-fuels, making it possible to recycle plant waste by converting lignocellulose (a non-food component, produced by plant walls) into sugars that are ethanol precursors.

In production processes based on lignocellulosic biomass (Figure 1.1), one of the crucial stages — and, above all, one of the most expensive — is the production of cellulases (enzymes) capable of making this conversion competitive. To improve this stage, we need to gain a clearer understanding of enzyme-producing microorganisms, such as *Trichoderma reesei*, a filamentous fungus.

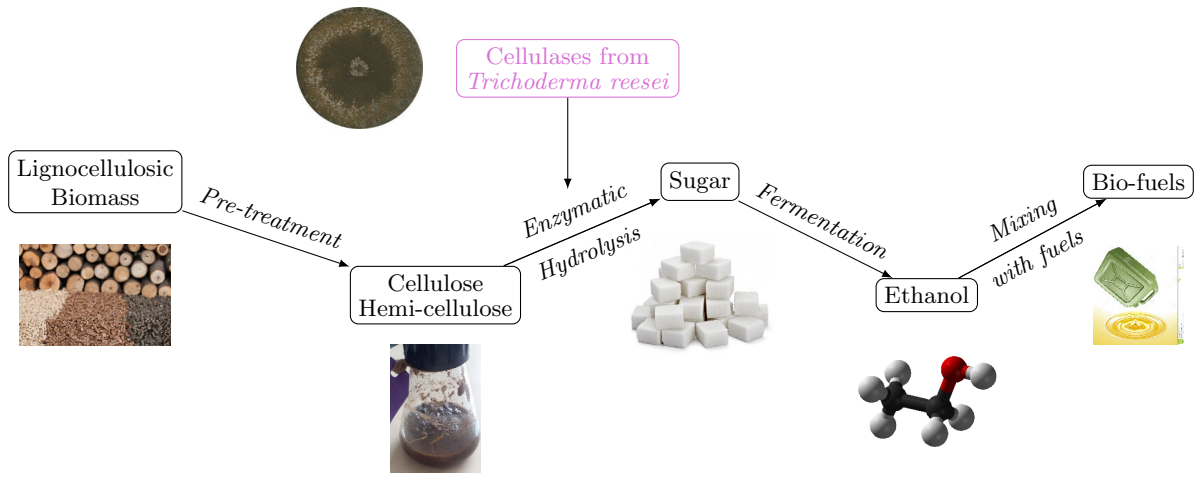


Figure 1.1 ~ Scheme of second generation bio-fuels process ~
The costliest step to be improved is highlighted in pink color.

Research protocols focusing on understanding living organisms have significantly been boosted by the emergence of what are known as "omic" technologies. Such data provide unprecedented access, on different scales, to fundamental biological mechanisms, thereby providing an abundance of complex information about how cells work. Analysis of the genome (DNA sequences),

the transcriptome (gene expression), or the metabolome (molecules produced by metabolism) are a few examples. Experiments of this type generate high volumes of data, offering a wealth of potential information but demanding cross-disciplinary skills, at the intersection of biotechnologies and algorithmic analysis development, for its effective integration and interpretation (Vert, 2013).

From this "omics" data, a large panel of bioinformatic tools is available. Specifically focusing on transcriptomic data allows us to better understand the genetic mechanisms yielding protein production. These data correspond — for a population of cells placed in various experimental conditions — to gene expression levels. They reflect, in a given experimental condition, which genes are actives and in which level. This kind of data require complex treatments, generally performed in an independent manner, encompassing various tasks at different scales: from the acquisition to the extraction of useful information. Briefly, classical bioinformatic workflows deal with image processing for acquiring data i.e. quantify the gene activity. Afterward, data normalization is performed in order to more rigorously compare gene expression level between experimental conditions. Statistical analysis is then usually carried out in order to detect genes having a particular behavior in at least one of the studied experimental conditions. Additional stages may then be performed in order to deeply explore the data. Notably, gene clustering allows us to group genes sharing similar gene expression levels across various experimental conditions. Grouped genes are expected to share similar genetic functions or to belong to a same biological pathway. Finally, constructing a graph encoding gene regulations is also a task of interest. In such graphs, nodes and edges are respectively derived from genes and their correlations or regulations. The resulting network is called a Gene Regulatory Network (GRN). Inferring GRNs from gene expression data is especially useful for sketching transcriptional regulatory pathways and helps to understand phenotype variations. However, these graphs, involving thousands of genes, are difficult to construct, visualize or analyze, especially when incorporating either experimental uncertainties or additional information retrieved from similar organisms. Despite the large number of available GRN inference methods, the problem remains challenging due to the under-determination in the space of possible solutions. Classical inference approaches rely on metric- or model-based strategies for assigning at each edge a weight reflecting the strength of the link between two genes. From these weights, the final curated network is then obtained after selecting only edges deemed relevant.

While all steps of such classical bioinformatic workflows (from data acquisition to data interpretation) are essential and cannot be neglected, in this thesis, our main focus was laid on the construction of GRNs. Although weights computation is a crucial step, the criterion defining which edges are relevant also reveals decisive. Our main contributions, summarized in the following section, rely on the establishment of novel criteria and the associated graph optimization methods for edge selection improvement in the context of the GRNs.

1.2 Contributions

Given all the pairwise gene regulation information available (i.e. edge weights), we propose to determine the presence of edges in the final GRN by adopting an energy optimization formulation. To refine inference results by restricting the space of possible solutions, additional constraints are incorporated into our models. Some constraints, reflecting either biological (information about gene interactions) or structural (information about node connectivity) *a priori*, have been considered. Different priors lead to different mathematical properties of the global cost function, for which various optimization strategies can be applied. Optimization strategies are inspired by recent graph optimization works in image processing and computer vision, where pixels and their connectivity are used to interpret images at a higher level. The post-processing network refinements we proposed led to a set of computational approaches named *BRANE* *** for "Biologically-Related A priori for Network Enhancement". For each of the propose methods, our contributions are threefold: *a priori*-based formulation, design of the optimization strategy and validation (numerical and/or biological) on benchmark datasets.

- ⋆ *BRANE Cut* (Chapter 4): it is our first edge selection strategy proposal for GRN refinement. The cost function we designed enforces a modular network arranged around central nodes, while a gene co-regulation *a priori* is used to constrain the space of possible solutions. When the co-regulation criterion we define is satisfied, a coupled edge inference is favored. The combination of this *a priori* allows us to formulate the problem as a minimum cut problem (also known as Graph Cuts in computer vision). Thanks to the duality between minimal cut and maximal flow, the proposed formulation can be solved using an efficient maximal flow algorithm pertaining to the class of discrete optimization algorithms. We also performed a numerical and biological evaluation of our proposed approach thanks to benchmark synthetic and real datasets. Comparisons performed with state-of-the-art methods are in favor of *BRANE Cut* (Pirayre *et al.*, 2015a).

- ⋆ *BRANE Relax* (Chapter 5): this second edge selection strategy is in the same vein as *BRANE Cut*, as the cost function we designed also enforces network modularity. Based on a biological postulate we additionally restrain the space of possible solutions by restricting the connectivity degree of particular nodes. The resulting discrete optimization problem is relaxed into a continuous one. A proximal splitting strategy yielding the use of a projected gradient algorithm is thus used for its resolution. Due to the potential high dimensionality of the problem, acceleration tricks relying on preconditioning and block coordinate strategy are complementary used. Performance of *BRANE Relax* is demonstrated through benchmark simulated datasets and shows improvement over state-of-the-art methods (Pirayre *et al.*, 2015b).

While *BRANE Cut* and *BRANE Relax* are exclusively focused on edge selection for GRN refinement, the last method we propose was thought to integrate gene clustering and GRN tasks in a jointly manner instead of an independent one. This approach was motivated by the drive to reduce the number of independent treatments classically performed on transcriptomic data, toward a tighter integration of elementary tasks in omics workflows.

- ⋆ *BRANE Clust* (Chapter 6): the cost function we designed allows us to jointly perform an edge selection and a gene clustering. In this formulation, we choose to promote the modular structure of the final network through the clustering. The resulting formulation relies on a discrete optimization problem for which an efficient alternating optimization procedure is proposed. An explicit solution can be computed for the edge selection sub-problem. After relaxing the gene clustering sub-problem, it can be solved *via* a random walker algorithm. Numerical performance of *BRANE Clust* was assessed on synthetic and real benchmark datasets. Significant improvements over state-of-the-art methods are also demonstrated. Biological relevance of both inferred GRN and gene clustering is also evaluated (Pirayre *et al.*, 2018a) .

Although this thesis was focused on the development of generic GRN inference methods, a complete bioinformatic study — from experimental design choice to biological interpretation of the results — was performed on *in-house* transcriptomic data regarding the fungus *Trichoderma reesei*. In addition to confirming established knowledge and to providing new insights on the genetic mechanisms engaged during the cellulase production, this bioinformatic study was used as a real case study for *BRANE Cut* use and blind validation without reference. Some applied results from our endeavor are disseminated in Poggi-Parodi *et al.* (2014); Pirayre *et al.* (2018b).

In a ramification of this thesis (Chapter 7), we extend our vision to more generic graph-based problems, not necessarily for GRN inference but keeping in mind forthcoming adaptations to biological purposes. We throw in *HOGMep*, a Bayesian approach developed for joint reconstruction and clustering on multi-component data. A Higher Order Graphical Model (HOGM) is employed on latent label variables for clustering or classification. In addition, a Multivariate Exponential Power (MEP) prior is opted for the signal in a given class. An efficient Variational Bayesian Approximation (VBA) was developed to solve the associated problem. In this preliminary work, we firstly demonstrate the performance of *HOGMep* in an image deconvolution context, in terms of quality of reconstruction (pixel recovery) as well as quality of segmentation (pixel classification) from synthetic and benchmark color images. Initiatory venture into medical (and unstructured) data classification has also been undertaken, with dissemination in Pirayre *et al.* (2017).

1.3 Publications, communications and codes

1.3.1 International journal papers

- ⋆ D. Poggi-Parodi, F. Bidard, A. Pirayre, T. Portnoy. C. Blugeon, B. Seiboth, C. P. Kubicek, S. Le Crom and A. Margeot
  Kinetic transcriptome reveals an essentially intact induction system in a cellulase hyper-producer *Trichoderma reesei* strain
  Biotechnology for Biofuels, December 2014, 7:173.

- ⋆ A. Pirayre, C. Couprie, F. Bidard, L. Duval and J.-C. Pesquet
  BRANE Cut: Biologically-Related Apriori Network Enhancement with Graph

cuts for Gene Regulatory Network Inference
BMC Bioinformatics, December 2015, 16:369.

⋆ A. Pirayre, C. Couprie, L. Duval and J.-C. Pesquet
BRANE Clust: Cluster-Assisted Gene Regulatory Network Inference Refinement
IEEE/ACM Transactions on Computational Biology and Bioinformatics, May 2018, 15:3.

1.3.2 International conference papers

⋆ A. Pirayre, C. Couprie, L. Duval and J.-C. Pesquet
Discrete vs Continuous Optimization for Gene Regulatory Network Inference
In *Proceedings of the International Biomedical and Astronomical Signal Processing (BASP) Frontiers Workshop*, pp. 23, Villars-sur-Ollon, Switzerland, 25-30 January, 2015.

⋆ A. Pirayre, C. Couprie, L. Duval and J.-C. Pesquet
Fast Convex Optimization for Connectivity Enforcement for Gene Regulatory Network Inference
In *Proceedings of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*, pages 1002–1006, Brisbane, Australia, 19-24 April, 2015.

⋆ A. Pirayre, C. Couprie, L. Duval and J.-C. Pesquet
Graph Inference Enhancement with Clustering: Application to Gene Regulatory Network Reconstruction
In *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO 2015)*, pages 2406–2410 , Nice, France, 31 August - 4 September, 2015.

⋆ L. Duval, A. Pirayre, X. Ning and I. W. Selesnick
Suppression de ligne de base et débruitage de chromatogrammes par pénalisation asymétrique de positivité et dérivées parcimonieuses
In *Actes du 25th colloque GRETSI*, Lyon, France, 8-11 September, 2015.

⋆ A. Pirayre, D. Ivanoff, E. Jourdier, A. Margeot, L. Duval and F. Bidard
Growing *Trichoderma reesei* on a mix of carbon sources reveals links between development and cellulase production
In *29th Fungal Genetics Conference*, Pacific Grove, CA, USA, 14-19 March, 2017.

⋆ A. Pirayre, Y. Zheng, J.-C. Pesquet and L. Duval
HOGMep: Variational Bayes and Higher-Order Graphical Models Applied to Joint image Segmentation and Reconstruction
Accepted (May 2017) to *International Conference on Image Processing (ICIP 2017)*, Beijing, China, 17-20 September, 2017.

1.3.3 Other oral communications

⋆ A. Pirayre, C. Couprie, L. Duval and J.-C. Pesquet
Graph enhancement via clustering: application to Gene Regulatory Network

inference
GdR MaDICS – One-day Workshop on Emerging Trends in Clustering, Orléans, France, 12 June 2015.

⋆ A. Pirayre, C. Couprie, F. Bidard, L. Duval and J.-C. Pesquet
Incorporating Structural A Priori in Gene Regulatory Network Inference using Graph Cuts
International Workshop on Algorithmics, Bioinformatics and Statistics for NGS data analysis (ABS4NGS), Paris, France, 22-23 June 2015.

⋆ A. Pirayre, C. Couprie, F. Bidard, L. Duval and J.-C. Pesquet
BRANE Cut: integrating biological a priori in Gene Regulatory Network inference with Graph cuts
Statomique, Paris, France, 9 November 2015.

⋆ A. Pirayre, D. Ivanoff, E. Jourdier, A. Margeot, L. Duval, and F. Bidard
Growing *Trichoderma reesei* on a mix of carbon sources reveals links between development and cellulase production
1st Trichoderma *Workshop, Satellite Meeting of the 13th European Conference on Fungal Genetics*, Paris, France, 3 April 2016.

⋆ A. Pirayre, C. Couprie, L. Duval and J.-C. Pesquet
Gene Regulatory Network inference refinement using clustering
GdR ISIS – Apprentissage et/ou traitement du signal et des images sur graphes, Paris, France, 17 June 2016.

1.3.4 Upcoming communications, submitted and in progress

⋆ A. Pirayre, C. Couprie, F. Bidard, L. Duval and J.-C. Pesquet
BRANE Cut : optimisation de graphes avec *a priori* pour la sélection de gènes dans des réseaux de régulation génétique
Submitted (April 2017) to *colloque GRETSI*, Juan-les-Pins, France, 5-8 September, 2017.

⋆ A. Pirayre, D. Ivanoff, L. Duval, C. Blugeon, C. Firmo, S. Perrin, E. Jourdier, A. Margeot and F. Bidard
Growing *Trichoderma reseei* on a mix of carbon sources suggests links between development and cellulase production
Submitted (May 2017) to *BMC Genomics*.

⋆ Y. Zheng, A. Pirayre, L. Duval and J.-C. Pesquet
Joint image and graph recovery and segmentation with variational Bayes and higher-order graphical models (HOGMep)
Submitted (May 2017) to *IEEE Transactions on Computational Imaging*.

1.3.5 Available software

⋆ *BRANE Cut*: http://www-syscom.univ-mlv.fr/~pirayre/Codes-GRN-BRANE-cut.html

- ⋆ *BRANE Clust*: http://www-syscom.univ-mlv.fr/~pirayre/Codes-GRN-BRANE-clust.html

1.3.6 Miscellaneous

- ⋆ Best Poster Presentation Runner-Up Award
  At *European Student Council Symposium (ESCS'2014)*, Strasbourg, France, 6 September 2014.
- ⋆ Selection for the final contest 3MT (3 Minutes Thesis), top 10 among 24
  At *23th European Signal Processing Conference (EUSIPCO 2015)*, Nice, France, 4 September 2015.
- ⋆ Selection for the 3 minutes thesis presentation https://www.youtube.com/watch?v=ZUQj9YMPdVU
  At *Yves Chauvin thesis award ceremony, IFP Energies nouvelles*, Rueil-Malmaison, France, 25 November 2015.

1.4 Outlines

This thesis is divided into 8 chapters. Following this introduction, Chapter 2 is devoted to an introductory part to bioinformatics with some recalls concerning biological notions and experimental processes for data acquisition. While not the main scope of this thesis, classical preliminary bioinformatic treatments are presented as they are ineluctable and provide some food for thought in perspectives. Chapter 3 is dedicated to a review of GRN inference methods and the strategy used to evaluate the developed ones, without omitting the presentation of mathematical tools used in this thesis. Chapters 4 to 6 are devoted to our software suite including *BRANE Cut*, *BRANE Relax* and *BRANE Clust*. In each chapter, chosen *a priori*, variational formulation and optimization strategy are detailed in addition to the assessment on benchmark datasets. In Chapter 7, inverse problems and Bayesian framework are introduced in a preamble of the description and evaluation of *HOGMep* in both an image processing and biological context. Finally, conclusions and perspectives are draw in Chapter 8.

2 Methodology

"L'esprit scientifique nous interdit d'avoir une opinion sur des questions que nous ne comprenons pas, sur des questions que nous ne savons pas formuler clairement. Avant tout, il faut savoir poser des problèmes."

Gaston Bachelard

This chapter is dedicated to the description of the workflow for dealing with transcriptomic data to infer gene regulatory networks and to discover the main actors responsible for protein production. We firstly recall some biological notions, necessary to understand the gene regulatory network inference problem. We then expose experimental principles to generate transcriptomic data from DNA microarray or RNA-seq experiments. Normalization and gene selection tasks are detailed before the introduction of gene regulatory network (GRN) concepts. Aspects of GRNs post-processing for network inference enhancement and analysis are also mentioned.

Contents

2.1 Biological prerequisites

A cell phenotype corresponds to an observable characteristic which is driven by the production of some specific proteins, itself driven by the expression of related genes. While some genes are expressed in a constitutive manner, some others depend on external and internal *stimuli*. This adaptation suggests the presence of gene expression regulatory mechanisms. Before comprehending protein production mechanisms related to a specific phenotype, it is necessary to understand protein origin in cells.

In molecular biology, the central dogma, as well a recurrent controversy (Crick, 1970; Schreiber, 2005; Stearns, 2010), can be formulated as: *one gene, one protein*. In the genome, a gene is defined — *sensu stricto* — as a DNA fragment carrying the instructions for making a protein. This meaningful information is encoded *via* a specific order of the nucleic bases A, T, C, G: it is the coding sequence which will be transcribed. In addition, a gene is also composed of a promoter containing an initiation sequence as well as regulatory sequences (enhancers and silencers). The promoter is located upstream to the coding sequence. Finally, at the end of the coding sequence, a terminator is found.

When gene expression is promoted, the coding sequence is transcribed into a messenger RNA (mRNA) by an enzyme named RNA polymerase. Except for the nucleic base T, which is replaced by the nucleic base U, the mRNA conserves the same sequence of nucleic bases as the corresponding gene. The mRNA, after a maturation step, is translated into a polymer of amino acids thanks to ribosomes. The synthesized polymer corresponds to the protein and its amino acid sequence is dictated by the sequence of nucleic bases of the mRNA. Figure 2.1 illustrates the protein synthesis process.

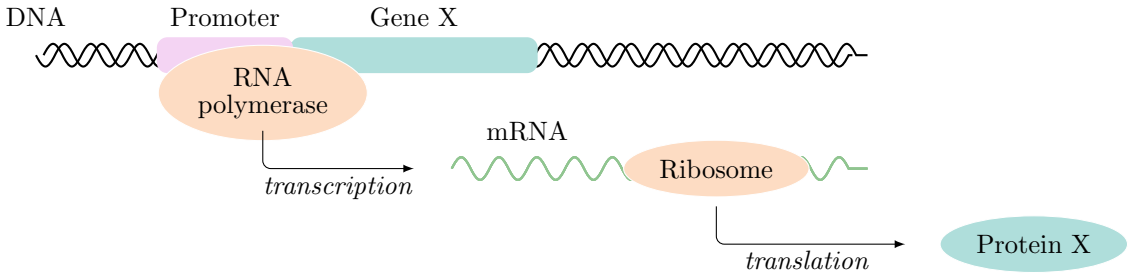


Figure 2.1 ~ Protein synthesis mechanism ~

Hence, a protein is present in a cell, as well as the corresponding mRNA, if its gene is activated. It is thus obvious that a dependence or association exists between protein production and gene expression regulation. We now explain some bases for gene expression regulation. The main regulatory mechanism involves the action of specific proteins called transcription factors (TFs). They can act alone or in association with other proteins in a complex. They recognize specific sequences (enhancers or silencers) located in the promoter of the genes that they regulate. TFs are responsible for two types of antagonist actions and can be:

- ⋆ activators: they increase the gene expression level. Activators are attached to enhancer sequences and promote the recruitment of the RNA polymerase.
- ⋆ repressors: they decrease the gene expression level. Repressors are attached to silencer sequences and block the recruitment of the RNA polymerase.

A same transcription factor may behave as an activator for one gene and as a repressor for an other gene. In addition, two mains complemental regulation strategies exist to control gene expression: epigenetic regulations, which are not directly related to a DNA sequence and post-transcriptional regulations which activate or inactivate a translated protein. These complex gene expression regulatory systems, which are all interdependent, make the discovery of gene regulatory pathways difficult. The integration of all these regulatory systems is discussed in Section 8.2, for further perspectives. Even if the regulation by TFs is only a part of the gene regulation, its knowledge is crucial to understand how proteins are produced.

When we are interested by the production of proteins (cellulases, for instance), discovering the regulation of corresponding genes is crucial. At the first scale, it is necessary to identify their direct TFs. The behavior (activator or repressor) of the identified TFs is also an essential information to be discovered. But, TFs acting in cascade, the identification of actors regulating these direct TFs is also needed, etc. This scheme results in a pathway and at the scale of several proteins, all the pathways generate a network called Gene Regulatory Network (GRN). In this present work on GRN inference, only TFs (repressors and/or activators) are specifically taken into account. Even for scarcely known organisms and strains, as it is the case for *Trichoderma reesei*, partial TF information is often available.

Unfortunately, gene regulatory mechanisms, with the actual technologies, cannot be directly observed. Biological experiments, *in silico* models and knowledge databases complemented by mathematical tools are thus necessary to discover and establish gene regulatory pathways. We now explain what transcriptomic data are and how to generate them (Section 2.2) before to briefly describe bioinformatic processes and workflows handling them (Sections 2.3 and 2.4) toward the GRN and pathways discovery finality.

2.2 Data acquisition and collections

The transcriptome refers to the set of all mRNA expressed in one or a population of cells, in a given experimental condition. Transcriptomic studies require as prerequisites to know where genes are located in the genome. In addition to qualitative information — *what genes are expressed?* — a transcriptomic study provides quantitative information — *in which levels?* In transcriptomic, the main postulate suggests that the amount of mRNA reflects the gene activation level and thus the amount of proteins in the studied condition. Hence, producing a set of transcriptomic studies in different experimental conditions allows us to obtain information on condition-dependent gene expression. Due to methodological limitations in transcriptomic data acquisition, comparisons between genes for a given condition cannot be performed. However, expressions over various conditions, for a given gene, may be compared. For instance, it is

possible to detect that gene X is more expressed in condition 1 than in condition 2. This is what we call a differential expression analysis. From transcriptomic data and differential expressions, it may thus be possible to infer gene-gene relationships reflecting regulatory mechanisms. Two main approaches produce transcriptomic data: DNA microarrays and, more recently with the advance of high-throughput sequencing, RNA-seq experiments.

2.2.1 DNA microarray principles and data

Several DNA microarray designs exist depending on the underlying biological question. In this work, we focus on the two-channel microarray of the Agilent platform used to produce in-house data on *Trichoderma reesei*. The SurePrint Technology® developed by Agilent is the most optimized technique. The popular Affymetrix platform relies on a similar principle. Agilent microarrays are conceived for differential analysis in gene expression. Assuming that we have the expression level for all genes in a reference condition, a two-channel microarray indicates the level of under- or overexpression for the same set of genes in a different condition. Its principle is detailed in Figure 2.2.

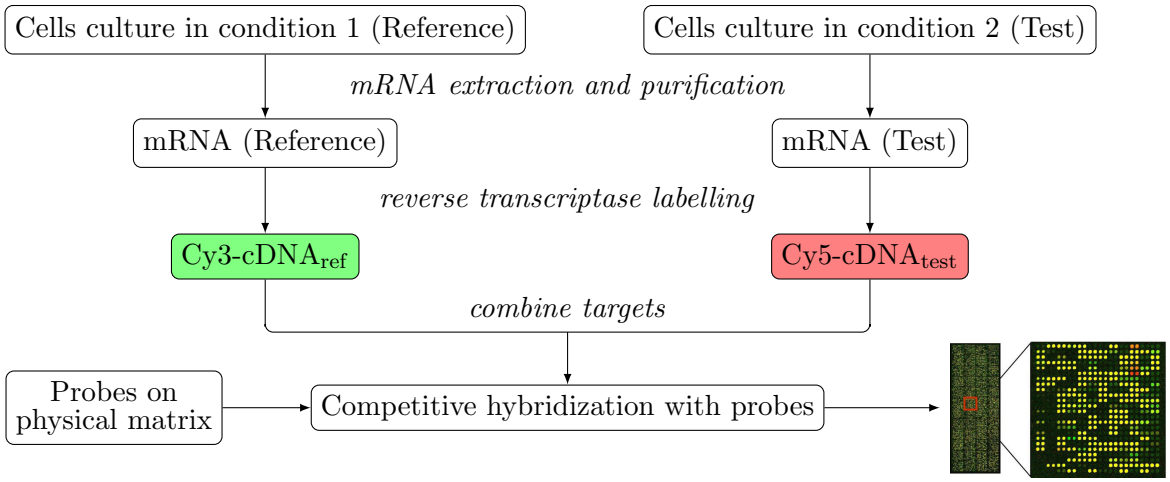


Figure 2.2 ~ Diagram of the principle of two-channel microarray technology ~

~ *Microarray preparation* ~ A microarray (or chip) is a physical matrix on which small DNA fragments, called probes (or oligonucleotides), from a given organism are immobilized in a random manner, see Figure 2.3(a). Each probe is referenced by its position on the chip (spot) and its nucleic sequence. Each spot contains many copies of the same probe, to facilitate the final detection by fluorescence approach. Probes may come from whole genome (genomic probes) or specific regions i.e. genes (transcriptomic probes). The latter is frequently used for transcriptomic studies as probes matching with genes only are interesting.

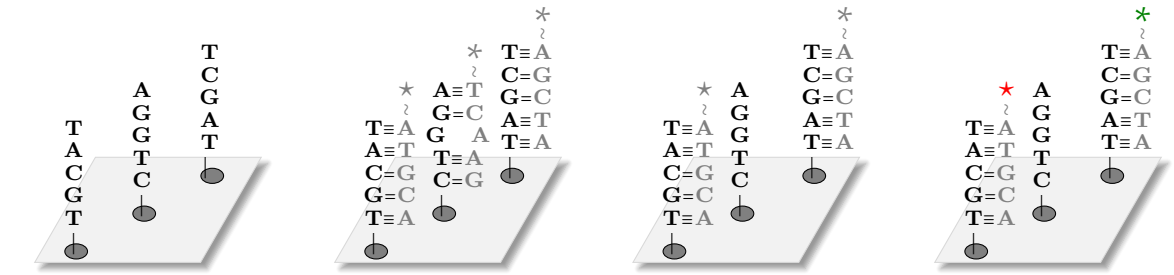
~ *Targets preparation* **~** Two cell cultures are necessary: a reference culture (time 0 h of a kinetic, for instance) and a test culture (24 h after the kinetic start, for instance). For each cell culture, mRNAs are extracted and purified. mRNAs have to be reverse transcribed to complementary DNA (cDNA) to make the hybridization with the probes possible. In the same time as cDNAs are synthesized, they are labeled by culture-dependent fluorochromes. The fluorochromes used are the cyanine molecules Cy3 and Cy5. Traditionally, the fluorochrome Cy3 is used for the reference culture and the Cy5 for the test culture. Separate solutions of Cy3-cDNA_{ref} and Cy5-$\text{cDNA}_{\text{test}}$ are then mixed yielding a unique solution of target fragments.

Microarray technology is based on the hybridization of two complementary single-stranded DNA fragments: probes and targets. Indeed, two complementary fragments naturally hybridize to constitute double-stranded DNA. The complementarity is base-dependent: A↔T and C↔G. *Now, back into the microarray context.* On the one hand, a chip with fixed single-stranded DNA probes is available. On the other hand, a solution of single-stranded DNA targets corresponding to a mixed Cy3-cDNA_{ref} and Cy5-$\text{cDNA}_{\text{test}}$ is also available. This solution is dropped off on the microarray containing probes and placed in a hybridization oven for one night. During this time, the microarray is spun in optimal conditions (pH, temperature, etc.) to favor hybridization between DNA probes and labeled cDNA targets, see Figure 2.3(b).

~ *Hybridization* **~** The hybridization is termed competitive, as a probe is complementary to both Cy3-cDNA_{ref} and Cy5-$\text{cDNA}_{\text{test}}$. For each probe matching a gene, the proportion of hybridized Cy3-cDNA_{ref} and Cy5-$\text{cDNA}_{\text{test}}$ reflects the amount of mRNAs and consequently the gene expression level in the reference and test culture condition, respectively. In other word, if the probe corresponding to a gene is more hybridized with Cy5-$\text{cDNA}_{\text{test}}$ than with Cy3-cDNA_{ref}, this implies that the gene is more expressed in the test than in the reference culture condition. In such a case, we say that the gene is overexpressed in the test condition. By analogy, we will say that a gene is underexpressed in the test condition if the hybridization level for Cy5-$\text{cDNA}_{\text{test}}$ is lower than for Cy3-cDNA_{ref}. Using this competitive hybridization, it is thus possible to detect differential gene expression between two conditions.

~ *Detection* **~** The proportions of hybridized targets are recovered using fluorescence of Cy3 and Cy5 fluorochromes, each of them depending on a target type (reference or test). After an overnight hybridization, all non-hybridized or badly-hybridized (non-specific) targets are firstly washed to avoid undesired fluorescence, see Figure 2.3(c). Then, the microarray is scanned and each spot is excited with a laser at respective wavelengths of 550 nm for Cy3 and 650 nm for Cy5. The emitted fluorescence (green for Cy3 and red for Cy5) is then collected *via* a photomultiplier (PMT) coupled to a confocal microscope. Two gray-scale images are obtained, one for each wavelength. The gray level reflects the emitted fluorescence intensity. Shades of gray are then converted to shades of green and red for the reference and the test image, respectively. Superposing the two colored images yields a unique false-colored image composed of spots from green to red, through yellow. This visualization allows us to observe differential gene expression.

- ⋆ Green-trend spot: Cy3-cDNA_{ref} was mostly hybridized. Corresponding genes are overex-



(a) Probe immobilization. *(b) Target hybridization.* *(c) Target washing.* *(d) Fluorescence detection.*

Figure 2.3 ~ PRINCIPLE OF THE HYBRIDIZATION IN A MICROARRAY ~
(a) DNA probes (base sequences in black) are immobilized on the microarray. (b) Reference and test targets (base sequences in gray) labeled by fluorochromes Cy3 (⋆) and Cy5 (⋆), respectively, are added for hybridization. (c) Non- and badly-hybridized fragments are washed. (d) Fluorescence detection is then performed by laser excitation.

pressed in the reference culture condition.

- ⋆ Red-trend spot: Cy5-cDNA$_{\text{test}}$ was mostly hybridized. Corresponding genes are overexpressed in the test culture condition.
- ⋆ Yellow-trend spot: Cy3-cDNA$_{\text{ref}}$ and Cy5-cDNA$_{\text{test}}$ were hybridized in a relative equal quantity.

Image processing is then used, including quality assessment and corrections, to quantify color intensities and thus differential gene expressions. For each spot (corresponding to a specific gene), green intensity and red intensity are obtained. The change of expression for a gene is then obtained by computed the red on green intensity ratio. As a side note, mathematical morphology has been a frequent tool for microarray data segmentation and quantification (Siddiqui *et al.*, 2002; Angulo and Serra, 2003). We refer to Kohane *et al.* (2003); Dougherty *et al.* (2005); Scherer (2009) for additional details on microarray signal and image processing.

This presented protocol is used to compare gene expression of only one test condition against a reference one. In a transcriptomic study, various test conditions are experimented, preferentially against the same reference condition. To limit fluorochrome-dependent biases, dye-swap experiments are usually performed. For a given *reference vs test* gene expression comparison, DNA of the reference culture is classically labeled with Cy3 fluorochromes while the DNA of the test culture is labeled with Cy5 fluorochromes. Dye-swap experiments consists in, at the same time, proceeding to the same *reference vs test* comparison while reversing the fluorochromes (reference DNA are labeled by Cy5 and test DNA by Cy3). In such a case, genetic materials are identical and experiments are called technical replicates.

In order to compare and deal with microarray data, normalizations are needed. Existing experimental biases corrections are presented in Section 2.3.1. Expression changes against the

reference and across the experimental conditions help us to detect regulatory relationships between genes as exposed in Section 2.4.

However, with the advance of high-throughput sequencing — and, in particular the Next Generation Sequencing (NGS) technology — a recent approach named RNA-seq, surpasses DNA microarrays for transcriptomic studies. We now detail the principles of RNA-seq data acquisition and highlight the main differences with DNA microarray data.

2.2.2 RNA-seq principles and data

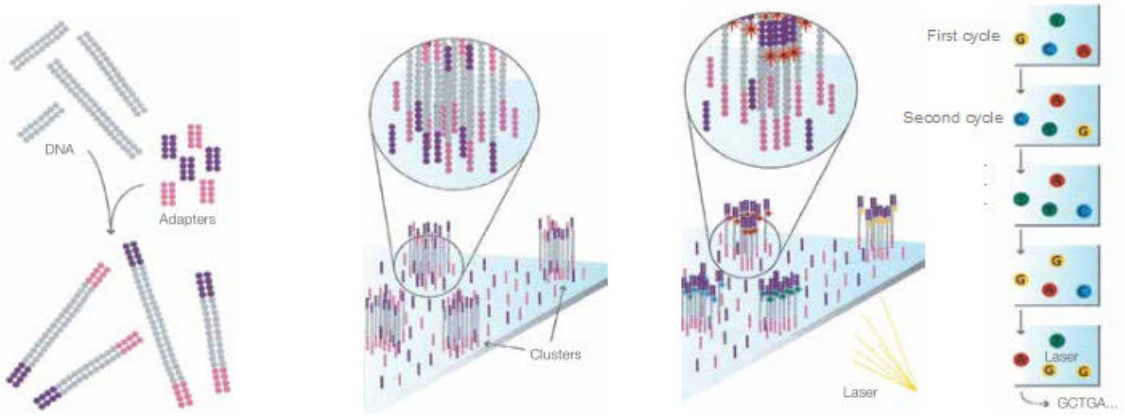
Next Generation Sequencing (NGS) is a relatively recent technology designed for whole genome sequencing i.e. to determine the linear order of nucleic bases A, T, C, G. But sequencing can also be used to quantify mRNA present in cells. It is called RNA-seq. Several RNA-seq technologies exist. We only expose the Illumina® Sequencing technology used to generate IFPEN data on *Trichoderma reesei*.

~ *Library preparation* **~** As we aim at quantifying gene expression levels, total RNA are firstly extracted from a cell culture of interest and mRNAs only are purified. They are then reverse transcribed into cDNAs. For practical reasons, cDNAs are fragmented to obtain smaller fragments of the same length (the size of the fragment conditions the technology to use) and an adapter is fixed at both cDNA extremities, see Figure 2.4(a). Here, we assume that the gene activity is reflected in the amount of mRNA, which is proportional to the amount of cDNA fragments.

~ *Cluster generation* **~** The cDNA-adapter complexes are then dropped off on a physical support called flow-cell. It contains complementary adapters to those ligated to the cDNAs, allowing the covalent fixation of the cDNAs to the flow-cell. Complexes are then amplified by Polymerization Chain Reaction (PCR). Each channel on the flow-cell is called a cluster and contains multiple copies of the same cDNA, see Figure 2.4(b). Hence, if a gene is highly expressed, a high number of clusters will contain cDNA fragments matching the corresponding gene.

~ *Sequencing and base calling* **~** Sequencing can now start. It is based on the natural DNA replication mechanism. An enzyme, called DNA polymerase, fixes the single-strand DNA and recognizes nucleic bases. At each base, the enzyme recruits the complementary base to synthesize the novel and complementary strand. In RNA-seq experiments, a DNA polymerase is used with fluorescent nucleotides, each type of nucleotide being associated to a fluorochrome. Sequencing is decoupled in cycles, where at each cycle, only one nucleotide is detected and identified. Hence, at the first sequencing cycle, DNA polymerase and labeled nucleotides are dropped off the flow-cell. In each cluster, the DNA polymerase uses the fluorescent nucleotide complementary to the first nucleotide of the cDNA. The flow-cell is scanned and a laser is used at the appropriate wavelengths to excite the four fluorochromes. In each cluster, thanks to the specificity between fluorescence color and nucleotide, the first base is detected and identified.

Sequencing and detection cycles are repeated until the end of the cDNA, see Figure 2.4(c). Image processing is needed for the base detection and identification step. Indeed, at each cycle, fluorochromes are excited using the appropriate wavelength and an image is taken. In this image, spots are present and correspond to the fluorescence in the cluster. Using image processing techniques, the fluorochrome giving the maximal fluorescence is identified in each cluster, and thus the incorporated nucleic base is identified. This process is called base calling. As we know where the cluster is located in the flow-cell, we can recover the sequence of the corresponding cDNA fragment. Such a sequence is called a sequence read or, simply, a read. Once all reads are obtained, additional processing is needed to quantify gene expression: quality assessment, read alignment, read counting, read count normalization.



(a) Library preparation. *(b) Cluster generation.* *(c) Sequencing and base calling.*

Figure 2.4 ~ ILLUSTRATION OF MAIN STEPS OF RNA-SEQ EXPERIMENTS. ~
Figure taken from Pub. No. 770-2007-002, Illumina documentation: `http://www.illumina.com/documents/products/techspotlights/techspotlight_sequencing.pdf`

Before the quantification of mRNAs to be able to conclude on the underlying gene expression levels, quality assessment has to be performed. For a given read, each sequenced base is evaluated using a Phred quality score. It measures the quality of the identification of the nucleobases generated by automated DNA sequencing. This score is logarithmically related to the probability of misidentification of a base. If reads are judged of good quality, aligning them to a reference genome is the next step, if an already-sequenced reference genome is available. We recall that a read is the sequence of a cDNA fragment corresponding to a part of an mRNA. The aligning step consists in mapping all sequenced cDNA fragments on the genome. Based on alignment results, the read count for the gene i, is obtained by identifying the number of sequenced cDNA fragments mapping the gene i. The read count reflects the absolute level of gene expression. This protocol is used to obtain each gene expression level for a given condition.

As for DNA microarrays, various conditions are tested. This requires normalization steps on read count data to compare experiments between them, see Section 2.3.1. Gene expression levels

obtained from read counts across various experimental conditions allow us to detect potential regulatory relationships between genes.

Why is RNA-seq preferred to microarray for transcriptomic studies? Several technical aspects are in favor of RNA-seq experiments. Firstly, microarrays are limited to known organisms as they require species- or transcript-specific probes, which is not the case in RNA-seq. In addition, supplemental detections can be made in RNA-seq such that novel transcripts, Single Nucleotide Polymorphisme (SNP)s, indels (small insertions or deletions) or isoforms. Secondly, unlike microarray, single or rare transcripts and weakly expressed genes can be detected in RNA-seq. This may be done by increasing the sequencing coverage depth[1]. Finally, not observable in RNA-seq, microarrays suffer from constrained dynamic range as gene expression measurements are limited by background and signal saturation. A key goal in transcriptomic studies is to detect condition-dependent changes in gene expression levels (detailed in Section 2.3.2). In the two-channel microarray technology, comparisons between conditions are defined in advance, through the experimental design, leading to relative gene expressions. On the contrary, RNA-seq technologies, providing absolute gene expressions, are thus more flexible and give additional degrees of freedom for the differential analysis.

DNA microarray and RNA-seq are both experimental techniques providing us the activity level of all genes. These data form the basis for the construction of gene regulatory networks. However, when dealing with quasi unknown organisms or a poor share of reliable information, we should first assess the trust one can place in network inference methods. Hence, before introducing the basics of GRN inference in Section 2.4, we evoke benchmark data to which a ground truth is associated.

2.2.3 Benchmark data: simulated and real compendium

When the aim is to develop new methods for inferring gene regulatory networks, a direct use of real data for which no or poor validation is available is not the best strategy. In this case, neither an objective validation nor a rigorous comparison with other GRN inference methods is possible. The lack of benchmark datasets with gold standards was resolved with the Dialogue on Reverse Engineering Assessment and Methods (DREAM) project. From a global viewpoint, the DREAM project assembles a community of researchers to promote open science in the field of biology and medicine. Indeed, they make available open and transparent data for rigorous and reproducible science. They cover a large panel of biological issues (Alzheimer's disease, prostate cancer, toxicogenetics, etc.) but also yet unsolved bioinformatics problems such as the estimation of model parameters, subclonal reconstruction algorithms or network inference among others. In the GRN inference context, the DREAM project propose three challenges DREAM3 (Prill *et al.*, 2010), DREAM4 (Marbach *et al.*, 2010) and DREAM5 (Marbach *et al.*, 2012). These specific challenges provide benchmark datasets as well as a standardized assessment methodology with ground truths to accurately compare GRN inference methods. Proposed performance metrics are discussed in Section 3.2.2. DREAM3 and DREAM4 challenges contain the same simulated

[1]Sequencing coverage depth refers to the number of times a nucleotide is read during the sequencing process.

data from *in silico* networks only while DREAM5 also present a compendium of real data in addition to simulated data. A detailed description of each dataset for the challenges DREAM4 and DREAM5 is provided in Section 3.2.1. Here, we focus on techniques to simulate gene expression data and give some words about real compendium datasets.

~ *Simulated benchmark datasets* ~ Simulated data are based on *in silico* networks, both generated by the tool GeneNetWeaver (GNW) (Schaffter *et al.*, 2011). A module extraction is firstly performed, from true biological networks (i.e. source networks), to obtain network structures. For this purpose, Marbach *et al.* (2009) propose to iteratively grow sub-networks from a given node until a fixed size such that the added nodes maximize a modularity index. This modularity Q is defined as the difference between the number of edges within the sub-network and the number of such edges in a randomized graph. Doing this, sub-networks resulting from the described module extraction are organized in a hierarchical modular structure, similarly to source networks. Once the structure is obtained, dynamic models are defined for gene regulation. Both transcription and translation processes are modeled through detailed kinetic models while molecular noise modeling is based on stochastic differential equations (Langevin equations). A supplemental experimental-like noise is added as a mixture of Gaussian and log-normal models seemingly observed in microarrays. These models are then used to generate gene expression data by simulating various biological experiments[2]: wild type, knockout, knockdown, dual knockdown or multifactorial. Each experiment can be simulated as steady-state or time-series.

From the generated *in silico* networks and gene expression data, GRN inference methods can be evaluated through objective performance metrics. Indeed, the *in silico* networks used to generate gene expression data are employed as ground truths for the assessment of predicted networks. Figure 2.5 illustrates the pipeline of benchmarking and assessment of GRN inference methods using GNW.

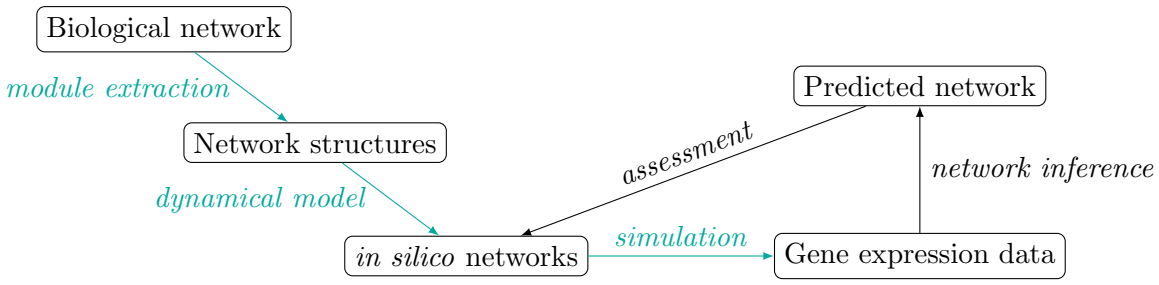


***Figure 2.5* ~** GeneNetWeaver pipeline **~**
Benchmarking and assessment of GRN inference methods. Green-labeled edges correspond to step specifically performed by GNW.

In addition to simulated gene expression data, the DREAM project also provide benchmark datasets coming from real experiments for which we briefly give some details.

[2]Definitions of the following biological experiments are given in the Glossary.

~ *Real compendium benchmark datasets* ~ In the field of the genetic of micro-organisms, very few species are sufficiently known to construct validated gene regulatory networks. Among the mostly studied species, *Escherichia coli* (*E. coli*) and *Saccharomyces cerevisae* (*S. cerevisae*) are used as models for prokaryote and eukaryote micro-organisms, respectively.

For these two species, various databases exist in which regulatory interactions can be extracted to construct a reference gene regulatory network used as ground truth. Specifically, for *E. coli*, the EcoCys (Keseler *et al.*, 2013) and RegulonDB (Gama-Castro *et al.*, 2011) databases contain manually curated known transcriptional interactions for which an evidence score is computed. In DREAM5, to construct the *E. coli* gold standard, the highest scored interactions are extracted from RegulonDB release 6.8 only. The gold standard for *S. cerevisae* comes from the study of MacIsaac *et al.* (2006) and has been chosen among a total of 16 gold standards derived from various studies (MacIsaac *et al.*, 2006; Hu *et al.*, 2007) and YEASTRACT database (Abdulrehman *et al.*, 2011). Note that, contrarily to *in silico* ground truths, such reference networks are not perfect. Indeed, even if we expect relatively few false positives, the number of false negatives is estimated with difficulties. Hence, objective performance metrics derived from these ground truths have to be considered with caution.

In addition to reference networks, compendia of published data are constructed for these two species. Data correspond to various microarray experiments coming from the same Affymetrix platform, similar to the Agilent platform presented in Section 2.2.1. They are downloaded from the Gene Expression Omnibus (GEO) (`http://www.ncbi.nlm.nih.gov/geo`). A normalization procedure — Robust Multichip Averaging (RMA) (Bolstad *et al.*, 2003) — is applied on these datasets to more rigorously cross-compare experiments. Moreover, in complement to the gene expression data, the identification of TFs is also performed thanks to Gene Ontology (GO) annotations.

In all experimental data, a pre-processing step is often inescapable. We now detail the most important pre-processing steps on gene expression data, either for two-channel DNA microarray or RNA-seq.

2.3 Gene expression pre-processing

Experimental data on gene expression try to reflect, at best, some biological reality. Unfortunately, due to technical and biological variability, gene expression data may be distorted. It is thus necessary to apply corrective treatments to overcome such biases. We firstly develop these data pre-processing techniques before evoking gene selection issues to be in more optimal condition for further analysis such as gene classification or gene network inference tasks.

2.3.1 Biases and normalization

Due to differences in experimental protocols, DNA microarray and RNA-seq data do not suffer from the same experimental biases. Consequently, normalization techniques have to be data-dependent, despite similar underlying biological assumptions. In the two following sections

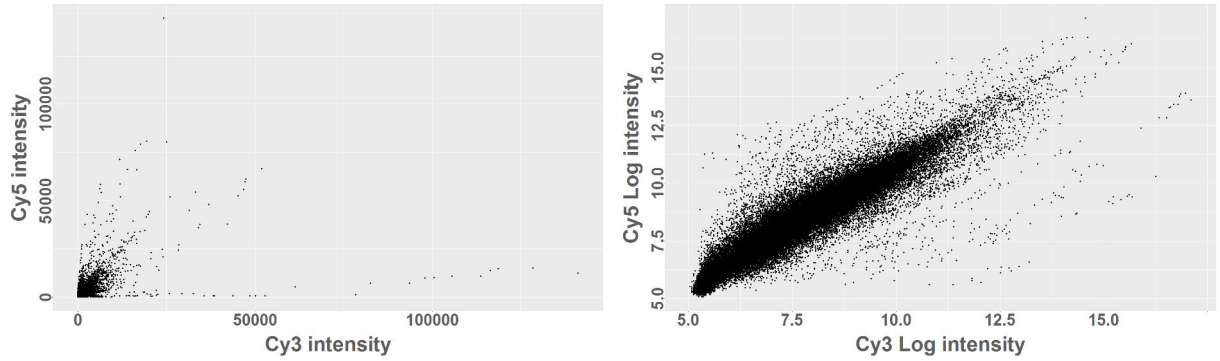
dedicated to the DNA microarray and RNA-seq data normalization, we deliberately detail commonly used normalization techniques. These details can appear — at first sight — superfluous in a GRN inference context. Nevertheless, it is important to keep in mind that gene expression data analysis calls in a complex pipeline based on a bunch of assumptions, that are not necessary transferable from one step to another. Normalization is one of the key step of this complex pipeline and cannot be neglected as it can influence GRN results (Lindöf and Olsson, 2003). Incidentally, one of the proposed perspectives of this thesis is to propose a novel normalization method that can be applied on both DNA microarray and RNA-seq data with a minimal number of hypothesis (Section 8.2).

~ *On DNA microarray data* ~ We recall that, for one experimental study, raw microarray data consist in a collection of green (Cy3) and red (Cy5) intensities for each spot (sometimes metonymically refered to as a gene). For a given spot $i \in \{1, \dots, N\}$, where N is the total number of spots, let $\mathtt{G}_i$ and $\mathtt{R}_i$ denote the green and the red intensity, respectively. As shown in Figure 2.6(a), intensity values are unequally spread over a large interval. We observe a large majority of genes for which red and green intensities are densely grouped on relatively small values. It is thus usual to take the binary logarithm of the intensities to reduce their scale of variation (Figure 2.6(b)). This transformation belongs to the family of variance-stabilizing transformations (Durbin *et al.*, 2002), with roots in works of Bartlett (1947) or Anscombe (1948). Several reasons can be evoked to justify the use of the binary logarithm (Reymond, 2004). Beyond an historical aspect, intensities measures are included from 0 to $2^{16} - 1$. In addition, the logarithm transformation takes the advantage to treat similarly over- and underexpressed genes. For instance, if a gene in the reference is twice more expressed, the intensity ratio equals 2 and the log-ratio 1. On the contrary, if the gene is twice less expressed in the reference, the intensity ratio equals 0.5 and the log-ratio -1.

In transcriptomic studies, the first main assumption dwells on the fact that most of genes would not see any change in their expression. Hence, by plotting bias-free red against green intensities for all genes, the slope should be 1. The second crucial assumption lies in the fact that the number of overexpressed and underexpressed genes tends to be similar. Based on this, true biological differences between the reference and the test condition can be detected above and below of the diagonal, in an equibalanced manner. Unfortunately, in addition to inherent biological variability, technical biases distort microarray data. These biases may be due, for instance, to a difference in the initial amount of mRNAs, in labeling efficiency of cDNAs, in laser excitation yielding variability in the emitted fluorescence, or in the amount of fixed probes on the chip. The impact of these disruptions may be observed on the red on green intensities plots (Figure 2.6) — usually called `RG`-plot .

Additional quantities may be defined from the binary logarithm of intensities (Dudoit *et al.*, 2002). For a given spot $i \in \{1, \dots, N\}$, where N is the total number of spots, we define the value $\mathtt{M}_i$ (log-ratio) as the binary logarithm of the intensity ratio:

$$\mathtt{M}_i = \log_2(\mathtt{R}_i) - \log_2(\mathtt{G}_i) = \log_2\left(\frac{\mathtt{R}_i}{\mathtt{G}_i}\right), \tag{2.1}$$



(a) *(b)*

Figure 2.6 ~ BINARY LOG TRANSFORMATION EFFECT ON RG-PLOT ~
(a) Distribution of red (Cy5) against green (Cy3) raw intensities. (b) Distribution of red (Cy5) against green (Cy3) intensities after binary log transformation. Intensities come from microarray data of the `NG14` *strain of* Trichoderma reesei *one hour after a lactose induction.*

and the $\mathtt{A}_i$ (mean average) value as the average log intensity:

$$\mathtt{A}_i = \frac{1}{2}\left(\log_2(\mathtt{R}_i) + \log_2(\mathtt{G}_i)\right) = \frac{1}{2}\log_2(\mathtt{R}_i\mathtt{G}_i). \tag{2.2}$$

From these two quantities $\mathtt{M} = \{\mathtt{M}_1, \ldots, \mathtt{M}_N\}$ and $\mathtt{A} = \{\mathtt{A}_1, \ldots, \mathtt{A}_N\}$, we usually visualize intensity-dependent ratios of raw microarray data through the `MA`-plot (Figure 2.7(a)). This plot is preferentially used to determine whether a normalization is needed. Based on the previous assumptions, bias-free `MA`-plot should show a majority of points on the y-axis (`M`) located at 0, independently of `A` values. Due to biases, this pattern is not recovered and a normalization is applied to be able to recover meaningful biological differences.

Quackenbush (2002); Yang *et al.* (2001) and Smyth and Speed (2003) provide an overview of microarray data normalization techniques. We may classify normalization approaches as follow:

- Within- or multiple-slides: the normalization applies on data coming from the same or different microarray(s).
- Paired-slides: the normalization applies on data coming from dye-swap experiments.

We only detail within- or multiple-slide normalization techniques and refers to Yang *et al.* (2001) for paired-slides normalization details (as rarely used in practice).

Global normalization This normalization relies on two assumptions: *i*) identical starting quantities of mRNAs are used for the reference and the test condition and *ii*) an approximately same number of marked reference and test cDNAs is hybridized. It results that these two quantities should be the same. In terms of normalization strategy, this boils down to searching a scale factor k such that $\mathtt{R}_i = k.\mathtt{G}_i$. Using the binary logarithm transformation, the normalization for each spot $i \in \{1, \ldots, N\}$ may be expressed as follow:

$$\log_2\left(\frac{\mathtt{R}_i}{\mathtt{G}_i}\right) \longrightarrow \log_2\left(\frac{\mathtt{R}_i}{k.\mathtt{G}_i}\right) = \log_2\left(\frac{\mathtt{R}_i}{\mathtt{G}_i}\right) - \log_2(k). \tag{2.3}$$

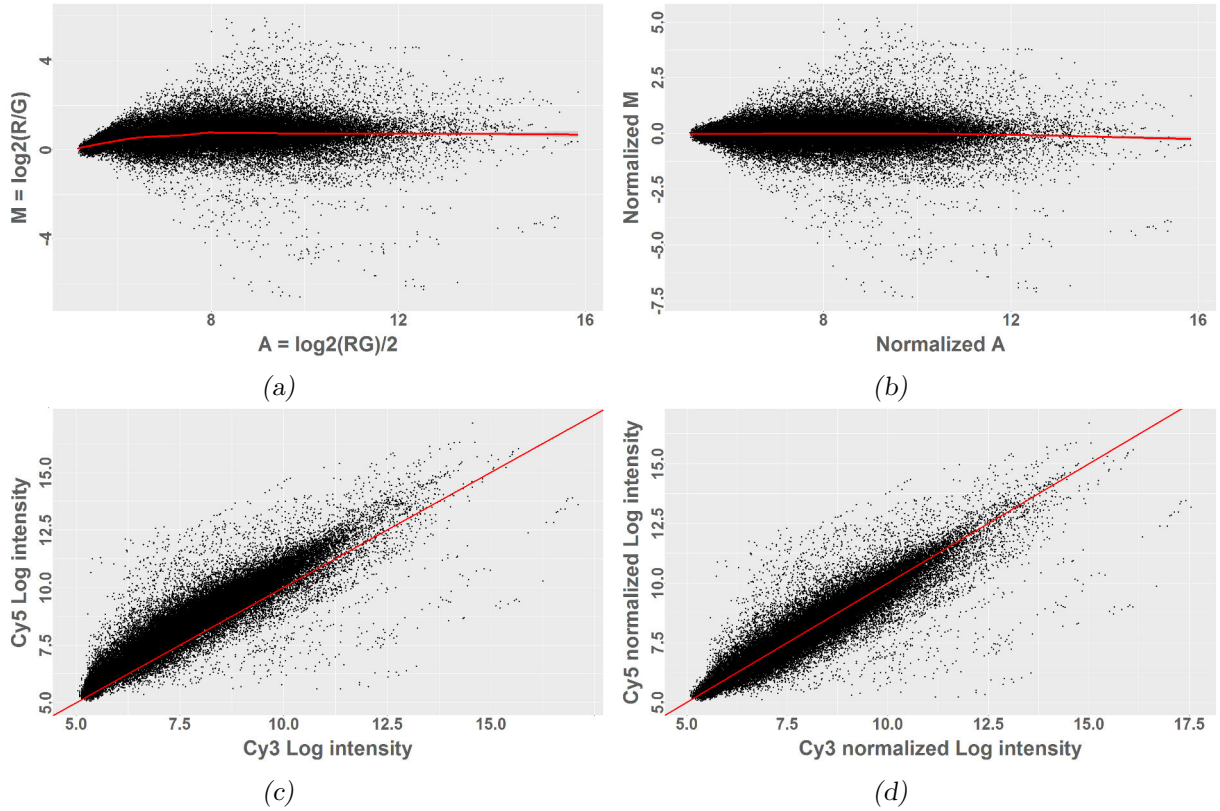


Figure 2.7 ~ LOWESS NORMALIZATION EFFECTS ON `MA`-PLOT AND `RG`-PLOT ~
`MA`-plot (a) and `RG`-plot (c) generated from raw intensities. `MA`-plot (b) and `RG`-plot (d) after LOWESS normalization. Red lines refers to the LOWESS curve obtained with (un)normalized data. Intensities come from microarray data of `NG14` strain of T. reesei *one hour after a lactose induction.*

> As we suppose that most genes would not see any change in their expression, the expected normalization aims at centering the distribution of the log-ratios (`M`) toward 0. Various strategies exist to define an appropriate log scale factor, but a suitable choice for the $\log_2(k)$ term is the median of the log-ratios. If we consider a quality weight on each spot, a weighted median of the log-ratios can thus be used as log scale factor. As this normalization only results in a global scale factor, the shape of point cloud remains the same.

However, this global normalization suffers from the intensity-dependent bias that may occur in the data. This bias is clearly visible on the `MA`-plot in Figure 2.7(a), where a deviation of 0 appears for low-intensity. This observation suggests that log-ratios have to be locally normalized. We now detail the most commonly used intensity-dependent normalization.

Intensity-dependent normalization The intensity-dependent normalization aims at locally centering the log-ratio distribution around 0. We thus look for an intensity-dependent

normalization scale factor, denoted by $l(\mathtt{A}_i)$, $i \in \{1, \ldots, N\}$. Yang *et al.* (2002b) use a LOcally WEighted Scatterplot Smoothing (LOWESS) (Cleveland, 1979) to perform the desired normalization. The normalization strategy boils down to searching the factor $l(\mathtt{A}_i)$ such that $\mathtt{R}_i = 2^{l(\mathtt{A}_i)}.\mathtt{G}_i$. Using the binary logarithm transformation, the normalization at each spot $i \in \{1, \ldots N\}$ may be expressed as follows:

$$\log_2\left(\frac{\mathtt{R}_i}{\mathtt{G}_i}\right) \longrightarrow \log_2\left(\frac{\mathtt{R}_i}{\mathtt{G}_i}\right) - l(\mathtt{A}_i), \tag{2.4}$$

where $l(\mathtt{A}_i)$ corresponds to the LOWESS estimate computed for the spot i. The LOWESS consists in multiple weighted least square regressions. Thanks to a bandwidth parameter, data are split into Q portions and a weighted regression is computed on each of them. Fitting results depend on the number Q of portions. The fewer the fraction, the smoother the fit. An optimization procedure to estimate the optimal bandwidth parameter is proposed by Berger *et al.* (2004). For a given estimation point, the weight function gives higher weights to closest points and the lowest to the most distant points. The tri-cubic function is traditionally chosen as such a weight function. The residual error may be computed and used to define additional robust weights. Using an iterative scheme, a final robust LOWESS estimate is obtained. This LOWESS curve is then used to correct an intensity baseline. Results using an intensity-dependent normalization based on LOWESS are displayed in the `MA`-plot of Figure 2.7(b) and in the `RG`-plot of Figure 2.7(c). The latter exhibits more centered intensities around the regression curve compared to the non-normalized `RG`-plot in Figure 2.7(b).

Similarly to LOWESS normalization, other smoothing approaches have been proposed, for instance: Splines Smoothing (SS) (Baird *et al.*, 2004; Workman *et al.*, 2002) and Wavelet Smoothing (WS) (Wang *et al.*, 2004). Nevertheless, no sensitive difference is observed between these approaches when they are compared. Additionally, Fujita *et al.* (2006) used Support Vector Regression (SVR) to normalize microarray data. Even if SVR normalization exhibits more robust results on the tested dataset, this approach is rarely used in practice. We refer to Park *et al.* (2003); Lim *et al.* (2007) and Fujita *et al.* (2006) for comparative studies of these normalization methods.

In addition to normalizing gene expressions for a given experiment, cross-experiment normalization may also be considered, as one of the finality of transcriptomic studies is to compare gene expression levels across experimental conditions.

Multiple-slides normalization In such approaches, each microarray is normalized separately according to one of the previous method. Hence, all normalized log-ratios for a given microarray are centered at 0. However, the variance of data generated by each microarray may be different and an additional step in the normalization is needed to unify the spread between experiments. A scaling factor for variance normalization may thus be applied to subdue this problem (Yang *et al.*, 2002b; Huber *et al.*, 2002). Nevertheless, this scaling normalization is not advised when the scale difference is small and its use has to be generally evaluated regarding the trade-off between its gain and a possible increase in variability.

Zien *et al.* (2001) propose a centralization method to directly normalize samples between them instead of treating them separately. This approach is based on the computation of a scaling factor for each sample obtained *via* maximum likelihood estimation.

Even if drawing a reliable conclusion remains difficult, intensity-dependent normalization traditionally produces better results. When a scaling normalization for variance stabilization is judiciously applied, better normalization results seem to be obtained.

However, the above normalization techniques cannot be applied on RNA-seq data. This is especially due to the fact that microarray normalizations are designed for relative gene expression based on red and green intensities. To that end, other normalization approaches have been developed to deal with RNA-seq. Their description follows.

~ *On RNA-seq data* ~ As mentioned, for a given experimental condition (sample) $j \in \{1, \ldots, S\}$, where S is the total number of samples, RNA-seq returns a count value — a read count — $R_{i,j}$ for each gene $i \in \{1, \ldots, G\}$, where G is the total number of genes. As a transcriptomic study implies various experimental conditions, read counts across these conditions have to be normalized for further analysis. Indeed, in addition to an inherent biological variability, technical biases require a normalization, which is context-dependent (Dillies *et al.*, 2013; Lin *et al.*, 2016).

For an inter-sample normalization, the main assumption is that only very few genes are Differentially Expressed (DE). This assumption implies that the read count distribution has to be the same across samples. Different strategies for distribution adjustment have been developed: Total Counts (TC), Upper Quartile (UQ) (Bullard *et al.*, 2010), Median (Med) (Dillies *et al.*, 2013), Quantile (Q) (Bolstad *et al.*, 2003) or Reads Per Kilobase per Million mapped reads (RPKM) (Mortazavi *et al.*, 2008). The latter also takes into account gene lengths in order to perform a gene-gene comparison for a given sample. Unfortunately, these approaches lead to unsatisfactory results. Another way to translate the low number of DE genes assumption lies in the fact that the total number of mapped reads (library size) has to be relatively close across the sample. Unfortunately, biases in the data lead to variability in library sizes and count distribution, as shown in Figures 2.8(a) and 2.8(b), respectively. The library size normalization consists in estimating a scaling factor which homogenizes all library sizes between samples to be normalized while preserving the dynamic of each samples. Trimmed Mean of `M`-values (TMM) proposed by Robinson and Oshlack (2010) — implemented in the R package *edgeR* (Robinson *et al.*, 2009) — and DESeq developed by Anders and Huber (2010) are the two mostly used normalization techniques for RNA-seq data. We thus choose to provide some details below.

TMM This inter-sample normalization requires to fix a sample as a reference sample and leave the others as test samples. Hence, we denote by $R_{i,j'}$ and $R_{i,j}$ the read counts of the gene i in the reference j' and test sample j, respectively. Similarly, $N_{j'}$ and N_j denote the total number of reads in the reference and the test sample respectively. For a given gene i and a given test sample j with respect to the reference sample j', Robinson and Oshlack (2010) define a log-ratio $\mathtt{M}_{i,j}^{(j')}$ and an absolute intensity $\mathtt{A}_{i,j}^{(j')}$, adapted from the microarray

framework:

$$\mathtt{M}_{i,j}^{(j')} = \log_2\left(\frac{R_{i,j}/N_j}{R_{i,j'}/N_{j'}}\right), \qquad \text{and} \qquad \mathtt{A}_{i,j}^{(j')} = \log_2\left(\frac{R_{i,j}}{N_j}.\frac{R_{i,j'}}{N_{j'}}\right). \tag{2.5}$$

Based on these new quantities, the scaling factor for the j-th sample can now be computed. A gene selection by double-trimming is firstly performed to remove the highest expressed genes (from the absolute intensity $\mathtt{A}$) and those exhibiting the highest log-ratios (from log-ratios $\mathtt{M}$). After trimming, the resulting set of genes is denoted by G^*. These genes should have the particularity to be moderately expressed and in the same manner in both the test and reference sample. Their log-ratios should ideally be equal to 1. In fact, they are not exactly equal to one, and this discrepancy is thus used to compute the scaling factor. Indeed, a weighted average on the remaining log-ratios gives us S_j, the scaling factor for the sample j:

$$S_j = \frac{\sum_{i\in G^*} w_{i,j}^{(j')} \mathtt{M}_{i,j}^{(j')}}{\sum_{i\in G^*} w_{i,j}^{(j')}}, \tag{2.6}$$

where $w_{i,j}^{(j')}$ are weights computed as the inverse of the approximate asymptotic variances:

$$w_{i,j}^{(j')} = \frac{N_j - R_{i,j}}{N_j R_{i,j}} + \frac{N_{j'} - R_{i,j'}}{N_{j'} R_{i,j'}}. \tag{2.7}$$

This approximation is obtained by the Delta Method detailed in Casella and Berger (2002, p. 240 sq.). Such weights take into account the fact that log-fold changes from genes with larger read counts have lower variance on the logarithm scale. This procedure is then repeated for each test sample $j \neq j'$. The normalized counts are obtained by dividing $R_{.,j}$, the raw counts for a given sample j, by the product of the initial library size N_j and the estimated scale factor S_j. By multiplying by one million, resulting normalized counts are called normalized Counts Per Million (CPM). Results of such normalization in terms of CPM library size and count distribution are displayed in Figures 2.8(a) and 2.8(b).

DESeq Anders and Huber (2010) propose another approach to estimate scale factors and to adjust for library size. Instead of choosing a reference sample, a virtual reference library is computed from raw counts. For each gene i, a central location estimator for read counts over the S samples (i.e. library), denoted by $\overline{R}_i$, is obtained by a geometric mean (Lawson and Lim, 2001). Then, intermediate read counts, denoted by $V_{i,j}$, are obtained by dividing read counts of gene i by $\overline{R}_i$, for all samples $j \in \{1, \ldots, S\}$. We note that, for computational reasons, the geometric mean is computed on non null elements only. We thus expect that genes having the same behavior in all conditions lead to an intermediate count V_i close to 1 in all samples. Due to technical biases, these reference counts may digress from 1. Hence, for each sample j, a scaling factor S_j can be obtained to adjust the library size between them. This factor S_j is obtained by computing the median over the G genes:

$$\forall j \in \{1, \ldots, S\}, \qquad S_j = \underset{i\in\{1,\ldots,G\}}{\text{median}} \{V_{i,j}\}. \tag{2.8}$$

Finally, in each sample j, read counts $R_{i,j}$ are divided by the corresponding scaling factor S_j, for all genes $i \in \{1, \ldots, G\}$. Results of the DESeq normalization are illustrated in Figures 2.8(e) and 2.8(f).

Results shown in Figure 2.8 display similar performances between TMM and DESeq normalization. In this illustration, normalizations are performed between the 36 conditions. A careful analysis shows that the library size was well scaled within biological replicates of a given condition, while a subtle bias remains between experimental conditions, especially for lactose condition at 24 h and 48 h (green and cyan bars). This remaining bias is probably due to a defective adjustment of extreme values, as observable in the normalized count distributions. Indeed, we remark a correct adjustment in terms of average (black line in boxes) and variance (box size) while the the extreme points (outside the central box) exhibit higher dispersion. This bias should not be forgotten in further analysis and interpretation. Despite these observations, TMM and DESeq normalizations are considered, to date, as the two best performers and are the most commonly used methods. Given the choice, DESeq normalization can be preferred, as it requires less parametrization. We note that in practice, normalization was performed on a set of conditions to be compared and not on the whole set of available conditions. Remind that the aforementioned methods often root on a large majority of genes keeping constant expression across conditions. Would the latter condition be violated, the use of normalization could even become harmful. It can be the case for specific studies where experimental conditions yield important cell changes across conditions. This happens for instance in a sporulation study: when a fungus takes its vegetative form, both its morphology and a large number of cellular functionality are affected, making the above assumption fragile.

One word on usable data (or genes). Every normalization method, either on microarray or RNA-seq data, aims at centering the log-ratio distribution around 0. By default, all available information is used. However, due to biological variability, taking into account all the data, corresponding to all genes, may be discussed. Indeed, normalization factors may also be computed from better selected subsets of data. For instance, particular genes called housekeeping genes (Eisenberg and Levanon, 2013) are expected to have the same activity (no significant changes in their expression levels) whatever the conditions — except for extreme stress conditions. We could thus be prompted to use their intensities only to compute normalization factors. Unfortunately, in practice, such housekeeping genes are badly identified and their availability is uncommon. Computing normalization on intensity data from housekeeping genes is thus rarely performed. Several alternative methods have been devised, at the closest to experiments. For instance in microarrays, one can use control spot. Two kinds of control spots exist and both have to be taken into account in the experimental design. The first strategy, called the spiked controls method, consists in using gene fragments coming from an organism different from the one being studied. These fragments, also called RNA spike-in, are fixed on the chip as probes (control spots) and are also injected in the same quantity in both the reference and the test mRNA samples. These control spots should produce the same red and green intensities and can be used for normalization. The second strategy, called titration series approach, uses the same probing gene introduced analogously in both reference and test mRNA solutions, at different concentra-

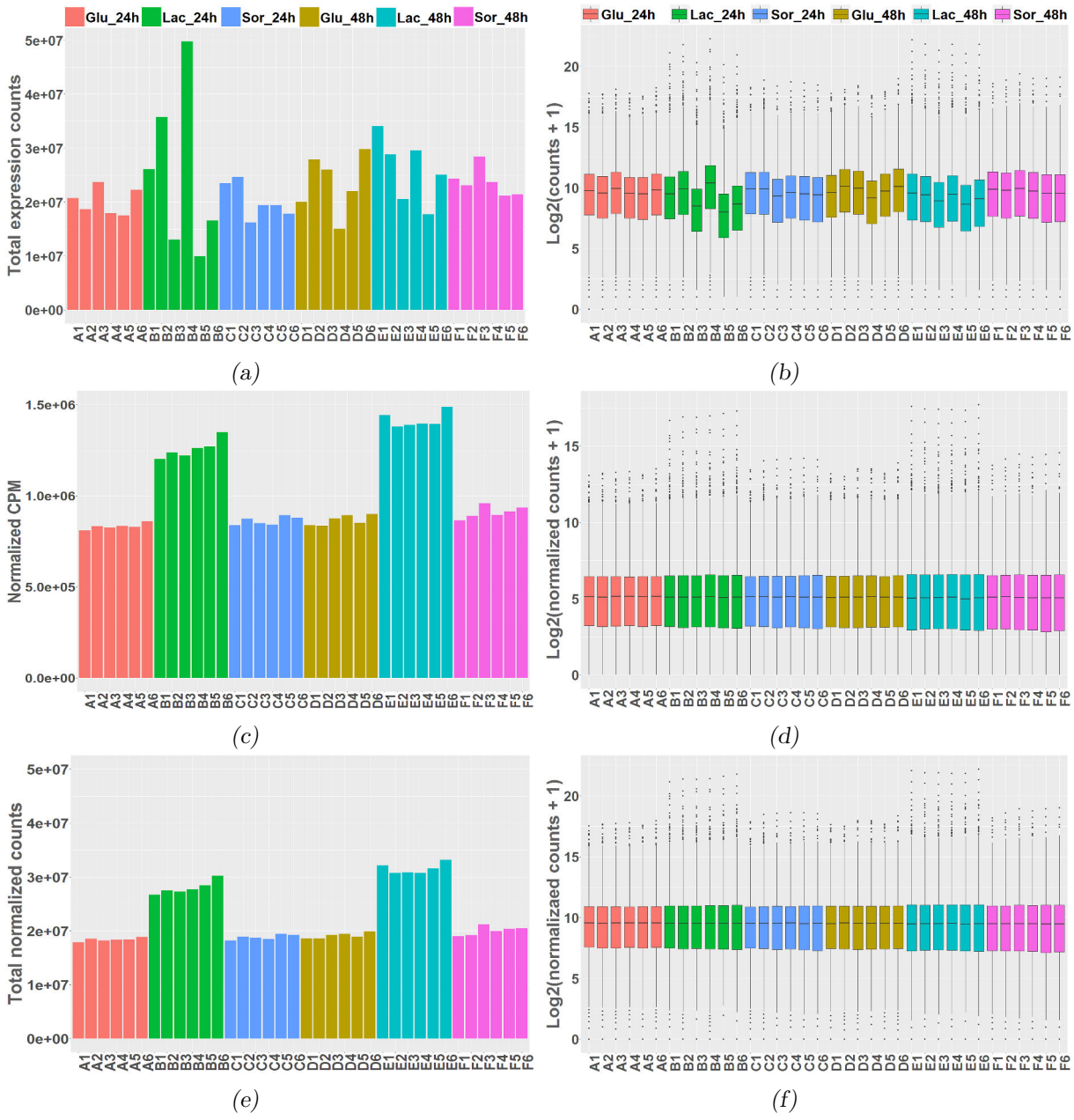


Figure 2.8 ~ Effects of TMM and DESeq normalizations. ~
Raw library sizes (a) and raw count distribution (b) from original data. Normalized library sizes after TMM (c) or DESeq (e) normalization. Results for TMM are given in normalized Counts Per Million (CPM). Normalized count distribution after TMM (c) or DESeq (e) normalization. Data obtained from RNA-seq experiments performed on 6 biological replicates of the `Rut-C30` *strain of* T. reesei *growing on different sugars (glucose, lactose or sorbitol) at* 24 h *or* 48 h.

tions. Regrettably, in practice, this approach is technically challenging and rarely used. We note that the TMM approach in Robinson and Oshlack (2010) somehow emulates the housekeeping gene concept by removing extreme data before computing normalization factors. An automatic detection of genes having a constant behavior in all conditions is discussed in Section 8.2.

The identification of gene expression changes with respect to various experimental conditions is one of the interest of a transcriptomic study. Once data normalization is performed, consistent gene expression comparison across conditions becomes possible. One now can detect which genes are impacted by a specific condition and how they are affected (under- or overexpression). This gene detection is called a differential expression (DE) analysis and allows to perform gene selection for further analysis e.g. clustering or gene regulatory network inference. We now present the main approaches — for microarray or RNA-seq — used to detect DE genes.

2.3.2 Differential expression and gene selection

A DE analysis aims at discovering genes that are differentially expressed between two conditions or more i.e. under- or overexpressed. This analysis suggests to detect genes whose behavior differs most between samples. Both the detection and analysis of DE genes can be an end *per se*. Nevertheless, they can also be used in order to restrict the set of genes for further analysis. This restriction makes sense as it decreases the disproportion between the number of genes and the number of observations — disproportion which can appear prejudicial in complemental analysis. Furthermore, working on DE genes only should focus results on singular behavior. Due to intrinsic differences between microarray and RNA-seq, data-specific normalization methods cohabit. From this section, we deal with — hopefully properly — normalized data. Note that, as for the normalization, the DE analysis plays a central role in the complex pipeline of gene expression data treatment. The profusion of DE analysis methods — involving always more additional assumptions — encourage us to, as for the normalization, propose a novel method as perspectives.

~ *On DNA microarray data* **~** Various approaches have been developed to detect DE genes according to experimental design. The most common statistical methods used are reviewed in Dudoit *et al.* (2002) and Cui and Churchill (2003).

To detect a change in gene expression between two conditions, an intuitive and basic way is to compute, for each gene i, a fold-change $\mathrm{FC_i}$, or its log transformed version $\log_2(\mathrm{FC_i})$. This fold-change often corresponds to the ratio $\frac{\mathtt{R}_i}{\mathtt{G}_i}$, where we recall that $\mathtt{R}_i$ and $\mathtt{G}_i$ denote the red and green intensities for the gene i, respectively. When biological replicates are available, the fold-change can be computed on averaged intensities. A global cut-off value is chosen from which ($\log_2$-)fold-changes are considered significant. More robust approaches based on Z-scores are initially employed to take into account both the mean and the standard deviation of the distribution of the ($\log_2$-)FC values across biological replicates. Significant DE genes are generally obtained for a confidence level of 95 %. However, these approaches are limited by an intensity-dependent effect observed on log-ratio variability (Chen *et al.*, 1997; Newton *et al.*, 2001). Yang *et al.* (2002a) propose to define intensity-dependent Z-scores for which mean and

standard deviation are locally computed.

When repetitions are available (samples corresponding to biological replicates), a Student's test (t-test) is classically preferred to evaluate the change of expression between two conditions (Callow *et al.*, 2000). For this purpose, the null hypothesis is defined as follows: gene expression levels are identical in the two tested conditions. For a given gene i, we recall that $\mathtt{M}_i = \log_2\left(\frac{\mathtt{R}_i}{\mathtt{G}_i}\right)$. The statistical test is thus expressed as:

$$t_i = \frac{\overline{\mathtt{M}_i}}{\mathrm{SE_i}} = \frac{\overline{\mathtt{M}_i}}{\sigma_i\sqrt{n_i}}, \tag{2.9}$$

where $\mathrm{SE_i}$ refers to the standard error for the gene i, $\overline{\mathtt{M}_i}$ and σ_i respectively denote the mean and the standard deviation of the $\mathtt{M}_i$ values across the n_i replicates. Unfortunately in practice, the number of replicates n_i is very low, resulting in instability in the gene-specific estimated standard deviation σ_i. To overcome this issue, assuming that the variance is homogeneous for different genes, a standard error SE across all genes can be computed, leading to a global t-test (Arfin *et al.*, 2000). However, the hypothesis of homogeneous variance across all genes may reveals erroneous. To take into account this heteroskedasticity, modified versions of the t-test have been developed. Notably, the regularized t-test, proposed by Baldi and Long (2001), adapts the denominator to both take into account the global σ and the gene-specific σ_i standard deviations. Their relative contributions are driven by a parameter v_0. In the Significance Analysis of Microarrays (SAM) approach developed by Tusher *et al.* (2001), the denominator is defined as the sum of the gene-specific standard error σ_i and a constant c which is usually defined as the 90-th percentile of the i-th standard error $\mathrm{SE_i}$. The B-statistic, developed in the work of Lönnstedt and Speed (2002), is defined as the logarithm of a ratio of probabilities. The latter ratio B is a posterior odds of differential expression as it corresponds to the probability for a gene to be differentially expressed divided by the probability for a gene not to be differentially expressed. A Bayesian framework, involving Gaussian and Gamma *priors* is used to compute the B-statistic for each gene. Smyth (2004) improves this previous statistic by reformulating the posterior odds, taking into account posterior residual standard deviation. This proposed moderated t-statistic, implemented in the R package *limma* (Smyth, 2005), provides good performance even on small numbers of replicates. It thus became a very commonly used procedure.

Once statistics for each gene are computed, their significance has to be evaluated. It is done by computing a p-value reflecting the probability to detect a false positive under a given distribution for the statistic. A gene is thus considered differentially expressed if its p-value is commonly lower than 1 % or 5 %. This result is obtained for one gene. As all genes are treated together, the problem resorts to multiple testing and p-values have to be adjusted. Befferroni (Dunn, 1959, 1961) or Benjamini-Hochberg (Benjamini and Hochberg, 1995) corrections are the two mostly used techniques to handle multiple testing issues and decrease the number of false positives. Note however that the traditional faith in p-values remains a debated topic (Wasserstein and Lazar, 2016).

Above methods are used to compare the gene expression level between two conditions. More

complex approaches are used to detect differentially expressed genes across more than two conditions. A commonly used approach is to perform an ANalysis Of Variance (ANOVA) (Kerr *et al.*, 2000). The microarray ANOVA model is defined from intensity data instead of dealing with log-ratios. Here, an F-test — which can be viewed as a generalization of the t-test — is obtained. The F-statistic is based on the comparison of the variation among replicates within and between conditions. Smyth (2004) proposed to fit a linear model to the expression data, log-ratios or log-intensities, for each gene. The resulting linear models can advantageously be adapted for a large panel of experimental designs. In addition, linear model fitting is combined with empirical Bayesian statistics previously evoked. The complete procedure can be entirely performed using the R package *limma* (Smyth, 2005).

Microarray experiments lead to intensity data and are thus treated as continuous measurements on which a log-normal distribution is assumed. However, RNA-seq experiments provide read counts: non-negative and discrete numbers. In this case, discrete distributions such as Poisson or Negative Binomial distributions are better suited. We now present DE gene detection dedicated to RNA-seq data.

~ *On RNA-seq data* ~ Overviews of the main approaches for DE analysis are provided in Oshlack *et al.* (2010) and Soneson and Delorenzi (2013). RNA-seq is a relatively novel method for which only few data validated processing tools and pipelines exist and have to be adjusted. A commonly assumed statement is that read counts generated by RNA-seq theoretically follow a binomial distribution. Let p be the probability that a read comes from a gene g. The binomial distribution is justified by the fact that, for a given gene g, the probability of obtaining that k reads over N come from the gene g is $\binom{N}{k} p^k (1-p)^{(N-k)}$. As the probability p is very small and N is large, the binomial distribution may be approximated by a Poisson distribution, with a unique parameter λ representing its mean. Unfortunately, the Poisson distribution is often too restrictive: mean and variance are assumed to be equal. Indeed, this strong assumption is rarely observed in practice, especially when biological replicates are available. In such a case, observed variance is significantly greater than the mean. This phenomenon is called overdispersion and has to be integrated for more reliable results. A negative binomial (NB) distribution is thus classically preferred to better take into account the variance (Robinson and Smyth, 2007). In such a case, both mean and variance (through the dispersion) have to be estimated for each gene and dispersion estimation is a crucial step in RNA-seq processing. From these estimated dispersions, statistical analysis are then performed in order to detect significant difference in gene expression levels. Common methods to detect differentially expressed genes from read counts are based on the Poisson (Auer and Doerge, 2011) or NB (Robinson *et al.*, 2009; Anders and Huber, 2010; Hardcastle and Kelly, 2010; Yanming *et al.*, 2011; Leng *et al.*, 2013) distributions.

In the Poisson-based framework, methods aim at estimating, for a gene i in a given condition j, the mean parameter $\lambda_{i,j}$ of the Poisson distribution from read counts only. In the TSPM method, Auer and Doerge (2011) define a statistical test to determine which genes have overdispersed counts. According to this test, genes are classified into two groups — genes with or without significant overdispersion — and the method used to detect DE differs according to the

group. For genes with overdispersion, they model gene expression with a quasi-likelihood (QL) approach which takes into account the overdispersion during mean estimation. Differentially expressed genes are then identified thanks to a likelihood ratio test statistic. For the remaining genes — without overdispersion — a standard likelihood approach is used.

When overdispersion is considered, read counts are mostly modeled by an NB distribution parametrized by the mean μ and the variance σ^2. Robinson and Smyth (2008) assume that mean and variance are related by $\sigma^2 = \mu(1+\phi\mu)$, where ϕ is the dispersion parameter. This ϕ parameter is assumed to be constant over experimental conditions and is estimated from the data *via* a conditional maximum likelihood approach for equally-sized libraries. A quantile adjustment is performed when library sizes differ. They improve this approach by estimating gene-specific dispersion parameters ϕ_i, $i \in \{1, \ldots, G\}$ using a weighted likelihood approach (Robinson and Smyth, 2007). This method is implemented in the R package *edgeR* (Robinson *et al.*, 2009). In Yanming *et al.* (2011), the relation between mean and variation are extended to $\sigma^2 = \mu(1+\phi\mu^{\alpha-1})$. Anders and Huber (2010) propose to estimate the dispersion using a local regression for the relation between mean and variance. This method is implemented in the R package *DESeq*. For these methods, an adapted exact test is used to statistically detect differentially expressed genes between two conditions. Linear models may be employed for more than two comparisons. As for microarray processing, statistical test are performed on each gene simultaneously and a p-value correction has to be applied to limit false positive detection (Dunn, 1959, 1961; Benjamini and Hochberg, 1995). We note that EdgeR and DESeq — which also encompass their respective normalization method presented in Section 2.3.1 — are the two most widely used methods for differential analysis. Other approaches, like baySeq (Hardcastle and Kelly, 2010) or EBSeq (Leng *et al.*, 2013) are also based on NB-distribution but use a Bayesian framework for dispersion estimation.

So what's next? Once DE genes are identified, a global analysis is generally performed in order to observe global transcriptomic changes: how many genes are DE? overexpressed? underexpressed? etc. They can also be specifically used for further analysis that aims at better understanding gene behaviors in specific experimental conditions, such as gene classification or gene network inference tasks. Working on DE genes derives from two main motivations. On one hand, we assume that cell phenotypic changes are mainly due to changes in gene expressions. Hence, genes tagged as non differentially expressed (NDE) are assumed to have no or a weak effect on the studied mechanisms. Removing them from further analysis is thus not nonsensical. On the other hand, due to the unfavorable data size and condition proportion — generally more than thousands of genes and less than 10 experimental conditions — performing gene classification or gene network inference is challenging and may lead to uninterpretable results. Using DE genes only is thus a suitable way to reduce the dimension of the data to be in more operational conditions for further analysis. Hence, after a differential analysis, only the normalized data of DE genes are used. These data correspond to the normalized log-ratios from microarray and normalized read counts from RNA-seq. It is also usual to compute log-ratios from normalized counts. The latter will be considered for the rest of this manuscript. Data can thus be gathered in a gene expression matrix $\mathbf{M} \in \mathbb{R}^{G\times S}$, where we recall that G is the number of genes and S the number of conditions (i.e. samples). The element $m_{i,j}$ corresponds to the log-ratio of the

gene i in the condition j. This gene expression matrix is used as input for the Gene Regulatory Network (GRN) inference task.

We now give a brief introduction to what a GRN is and how the graph framework can be employed. Section 3.1 is dedicated to related works on GRN.

2.4 Gene Regulatory Network (GRN) inference

As exposed in Section 2.1, gene expression leads to proteins. Some of these proteins have regulatory functions i.e. these proteins, called transcription factors (TFs), regulate the expression of other genes, denoted as $\overline{\text{TF}}$s. The action of TFs is not isolated and is integrated in a complex pathway. A toy example of such a regulatory mechanism is provided in Figure 2.9.

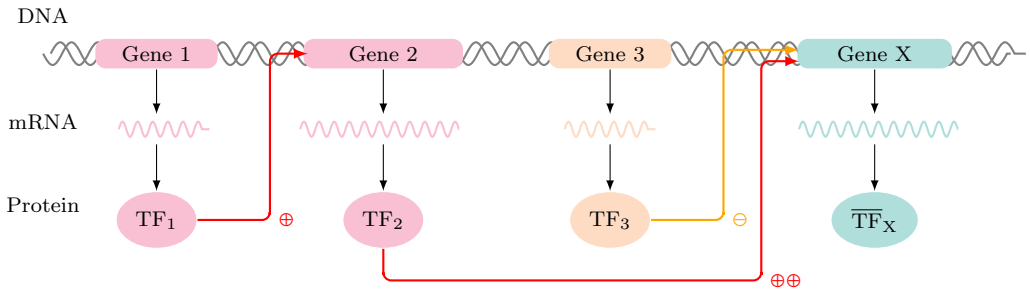


Figure 2.9 ~ Gene regulatory mechanism ~
Illustrated gene regulation involved transcription factors. Gene 1 is firstly transcribed and the resulting mRNA translated into the TF_1. *This* TF, *which is an activator, will activate the expression of gene 2, which in turn will be transcribed to obtain* TF_2. *In the same time, gene 3 is also active to produce* TF_3, *which is an inhibitor. Both the activator* TF_2 *and the inhibitor* TF_3 *act together to regulate the expression of the gene X coding for a* $\overline{\text{TF}}$. *The expression of the gene X is induced by* TF_2, *yielding the production of the* $\overline{\text{TF}}_X$, *but the presence of the repressor* TF_3 *decreases its maximal expression. This pathway is modeled as a graph in Figure 2.10.*

Graph structures unveil a suitable way to represent this regulatory pathway (Klamt *et al.*, 2009). A graph is composed of two objects: nodes (or vertices) and edges (or arcs), which tie nodes together. In the case of a gene network, nodes correspond to genes. To simplify explanations and notations in this manuscript, genes, mRNAs and proteins will be assimilated to the same entity and are put under the control of the gene. An edge between two nodes is built if there exists a biological relationship between the two corresponding genes. Gene regulatory networks specifically contain functional links reflecting causal interactions mainly between transcription factors and their targets genes. Figure 2.10 shows the corresponding graph encoding the regulatory mechanism displayed in Figure 2.9. GRN inference aims at recovering true regulatory links between genes from biological data such as transcriptomic data e.g. the gene expression matrix $\mathbf{M}$.

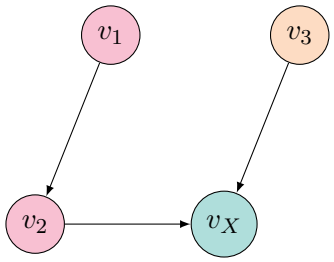


Figure 2.10 ~ GRAPH STRUCTURE ENCODING A GENE REGULATORY MECHANISM ~
Nodes correspond to genes and links between nodes to regulatory interactions derived from Figure 2.9. Pink and orange nodes represent TFs*: activator and repressor, respectively. The green node represents the protein of interest to be regulated.*

More formally, let $\mathcal{G}_\mathcal{V}$ be a complete unweighted and node-valued graph (Berge, 1973; Merris, 2000; Bondy and Murty, 2007). The set of nodes (corresponding to genes) is denoted by $\mathcal{V} = \{v_1, \ldots, v_G\}$, where G is the number of genes. We introduce $\mathbb{V} = \{1, \ldots, G\}$ as the set of node indices. The set $\mathcal{E}$ refers to the set of edges, corresponding to plausible interaction between genes. An edge between nodes i and j is labeled by $e_{i,j}$. We recall that transcriptomic data are gathered in the gene expression matrix $\mathbf{M} = [\mathbf{m}_1, \ldots, \mathbf{m}_G]^\top$, where, for all $i \in \mathbb{V}$, the vector $\mathbf{m}_i = [m_{i,1}, \ldots, m_{i,S}]$ reflects the expression profile of the gene i i.e. the set of log-ratios for the gene i over the S conditions. From these data, nodes of the graph $\mathcal{G}_\mathcal{V}$ can be multi-valued by the expression profiles i.e. node v_i is valued by the vector $\mathbf{m}_i$. The associated unweighted adjacency matrix[3] is denoted by $\mathbf{W}_\mathcal{V} = \mathbf{1}$, where $\mathbf{1}$ refers to a matrix of size $G \times G$ full of 1. From this multi-valued graph on nodes, the inference consists in recovering true regulatory links between genes. The resulting set of true links is denoted by $\mathcal{E}^*$ and the underlying graph $\mathcal{G}^*$. While some methods propose to directly infer the GRN $\mathcal{G}^*$ from $\mathcal{G}_\mathcal{V}$, some others require two steps. Firstly gene-gene interaction scores are computed leading to a gene-gene interaction matrix $\mathbf{W}_\mathcal{E} \in \mathbb{R}^{G \times G}$, where the element $\omega_{i,j}$ of $\mathbf{W}_\mathcal{E}$ is a weight reflecting the strength of the interaction between node i and j. Weights in $\mathbf{W}_\mathcal{E}$ are computed from expression profiles in $\mathbf{M}$. The gene-gene interaction matrix allows us to define the graph $\mathcal{G}_\mathcal{E}$ where nodes are non-valued while edges $e_{i,j}$ are weighted by the element $\omega_{i,j}$ of the matrix $\mathbf{W}_\mathcal{E}$. In such a case, the matrix $\mathbf{W}_\mathcal{E}$ defines the adjacency matrix of the graph $\mathcal{G}_\mathcal{E}$. As nodes and edges are the same in $\mathcal{G}_\mathcal{V}$ and $\mathcal{G}_\mathcal{E}$, we can use the same notation for their respective sets of nodes and edges. From the fully-connected and weighted network $\mathcal{G}_\mathcal{E}$, an edge selection is performed to recover $\mathcal{E}^*$ by retaining edges having relevant weights only, ideally corresponding to true regulatory relationships. This edge selection task is classically performed by removing all edges whose weights $\omega_{i,j}$ (possibly their absolute value) are lower than a threshold λ. Figure 2.11 illustrates the main steps on the exposed gene regulatory inference, on the toy example of Figure 2.9. To sum up, the graph $\mathcal{G}_\mathcal{V}$ encodes the gene expression data and can be directly used to recover $\mathcal{G}^*$, or used to define an intermediate graph $\mathcal{G}_\mathcal{E}$ to be pruned to find $\mathcal{G}^*$. An overview of GRN inference approaches is given in Section 3.1. For the rest of the manuscript, notations $\mathcal{G}_\mathcal{V}$ and $\mathcal{G}_\mathcal{E}$ will be confounded into a unique notation $\mathcal{G}$. Reference to $\mathcal{G}_\mathcal{V}$ will be made through the notion of (unweighted) node-value graph where $\mathbf{W} = \mathbf{1}$. In the same vein, reference to $\mathcal{G}_\mathcal{E}$ will be made through the notion of weighted edge-valued graph where $\mathbf{W} = f(\mathbf{M})$, where f is a function returning gene-gene interaction scores. We refer to Section 3.1 for an overview of weights computation methods encoding such a function.

[3] Matrix encoding the graph structure by setting elements to 1 when an edge is present in the graph and 0 otherwise.

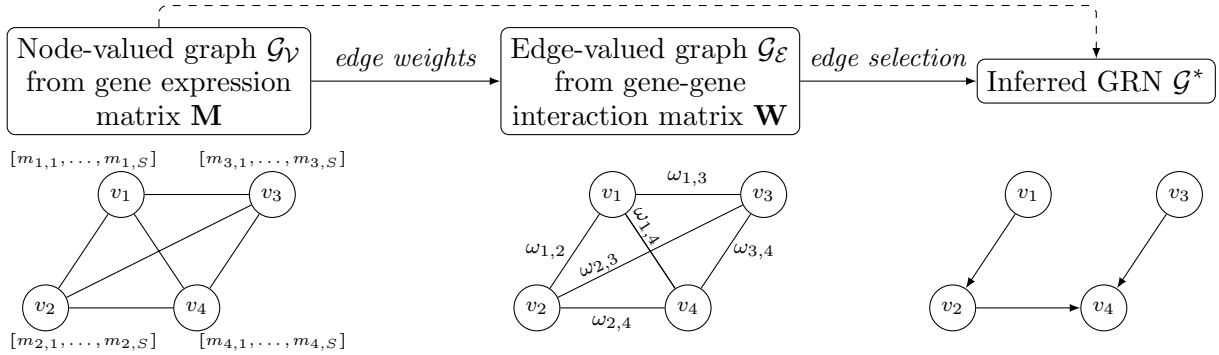


Figure 2.11 ~ Main steps of gene regulatory network inference ~

However, a large majority of methods fail to infer a GRN in a reliable manner and generally suffer from systematic prediction errors (Marbach *et al.*, 2010). The first one is the inference of links between two co-regulated target genes: a link between $\overline{\text{TF}}$s i and i' is added if genes i and i' are both regulated by the same TF j. These kinds of links are misinterpreted as co-regulation links while they reflect co-expression. They are thus unwanted in a GRN and have to be removed. The second one refers to indirect interactions occurring in an inferred regulatory cascade: a link between node i and k is added if the cascade $i \to j \to k$ is inferred. As presented in Section 3.1, some methods have been proposed to remove indirect links. Finally, the third classical prediction error lies on the difficulty to correctly infer combinatorial regulation i.e. a gene which is regulated by multiple TFs. In addition to these classical biases, Marbach *et al.* (2010) showed a poor overlap between inferred networks from a compendium of methods. Merging complemental GRNs gives higher performance and leads to a more interpretable network. However, this merging is not performed in practice due to its computational time cost. These limitations, in addition to the disproportion between the number of genes and the number of observations, could explain why — still at present — GRN inference remains an ill-posed, opened and unsolved problem.

What if the edge selection was seen as an optimization problem? While the computation of gene-gene interaction scores is a crucial step in the inference process, the edge selection step is also an essential task — though often neglected — to obtain biologically relevant results. As mentioned, classical selection results in a unique thresholding removing edges whose weights have a magnitude lower than a threshold λ. In this thesis, we propose to handle this edge selection issue *via* graph optimization. For this purpose, the classical thresholding can be expressed as a regularized optimization problem for which the explicit solution directly gives the set of edges having a weight higher than a threshold λ. Details regarding this approach are presented in Section 3.3.1. The main contributions of this thesis is to improve this classical edge selection by integrating biological and structural *a priori* in addition to favor high-weighted edges. Three novel optimization formulations have been proposed: *BRANE Cut*, *BRANE Relax* and *BRANE Clust*. They are presented in details in Chapters 4, 5 and 6, respectively.

Once GRNs are constructed, an additional treatment, sometimes referred as network post-processing, can be applied to analyze them. Post-processing on GRN may lead to different but complementary results such as, for instance, module detection with Weighted correlation network analysis (WGCNA) (Langfelder and Horvath, 2008). By modules, authors understand groups of genes that are highly connected. They are detected *via* unsupervised clustering and significance module scores are assigned to select biologically significant modules. Gene clustering from GRN is also proposed in Rapaport *et al.* (2007) where the GRN is used as *a priori*. They construct a classifier which groups predictor variables according to their neighborhood relations in the network. Differential network analysis may also be performed to compare GRNs and extract group-specific networks such as in Differential network analysis in genomics (DINGO) from Ha *et al.* (2015) or in Okawa *et al.* (2015). In a more biological approach, a set of tools, detailed in Section 3.2.2 can also be used as post-processing to detect new biological insights in the GRN. Hence, in addition to the GRN construction, these supplementary analyses are used to better understand regulatory pathways in cells. In a context of genetic engineering, these tools are useful to detect both TFs and their targets involved in the expression of proteins of interest. Figure 2.12 recaps the usual workflow of gene regulatory network use. This thesis is focused on the network inference part and more specifically the edge selection task. However, all the stages presented have been taken up in order to highlight and master key issues in gene network inference and propose more adaptive solutions to fix them.

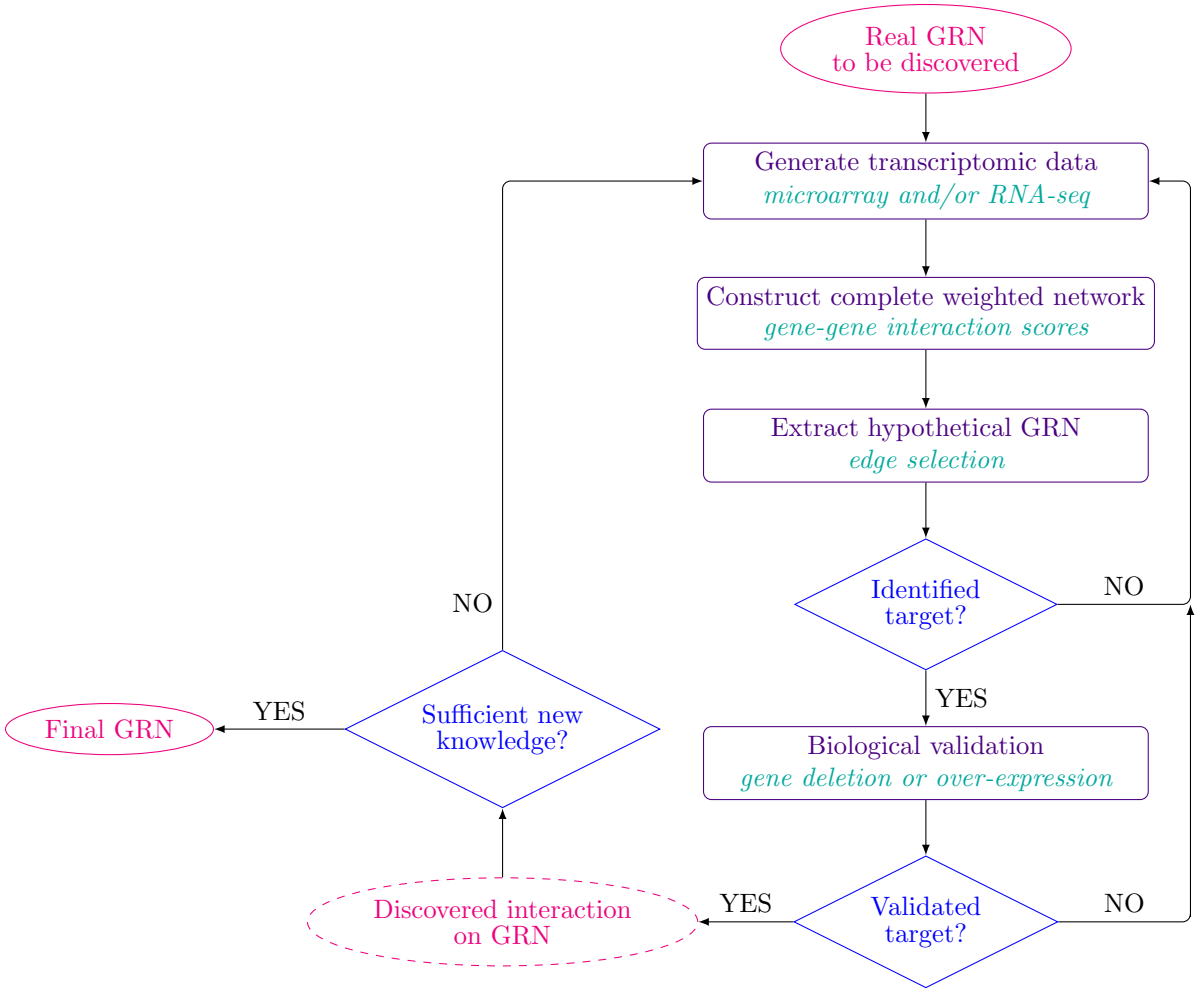


Figure 2.12 ~ SUMMING-UP OF THE MAIN STAGES OF GENETIC ENGINEERING ~
Discovering an unknown GRN requires the acquisition of transcriptomic data, classically from microarray or RNA-seq experiments. From these data, a complete weighted network is built by assigning to each edge $e_{i,j}$ a weight reflecting the strength of the interaction between genes i and j. Thanks to an edge selection step, an hypothetical GRN is extracted, from which candidate genes for the studied mechanism can be supposed. When gene candidates are identified, biological experiments are carried out to validate them, allowing to complete the unknown GRN. These novel interactions can be an end per se. *However, if we judge the additional knowledge insufficient, the complete procedure can be repeated, up to obtain a sufficiently complete final GRN.*

3 An overview of related works in GRN inference

"I believe the day must come when the biologist will — without being a mathematician — not hesitate to use mathematical analysis when he requires it."

Karl Pearson

In this chapter, we focus on the gene regulatory network (GRN) inference problem. A detailed overview of related works on this subject is firstly presented. In addition, advantages and limitations of the current state-of-the-art methods are discussed. We then expose the strategy, from data to databases, used to validate and compare our proposed methods with state-of-the-art approaches. Finally, some mathematical basics and optimization tools employed in the developed methods are given.

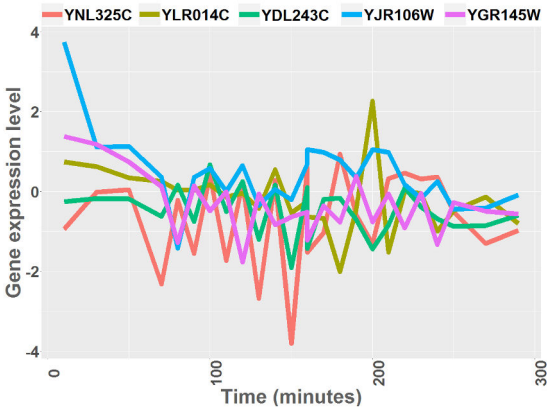
Contents

3.1 GRN inference methods

This section is dedicated to a detailed overview of Gene Regulatory Network (GRN) inference methods. Let us recall some notations. The common input of the methods is the gene expression matrix $\mathbf{M} \in \mathbb{R}^{G \times S}$ gathering, for every gene $i \in \{1, \dots, G\}$, the expression profile $\mathbf{m}_i$ of length S, where S is the number of experimental conditions. Figure 3.1 illustrates an excerpt from this kind of data.

$$\mathbf{M} = \overbrace{\begin{pmatrix} -0.948 & -0.013 & \dots & -1.308 & -0.977 \\ 0.737 & 0.619 & \dots & -0.141 & -0.803 \\ -0.253 & -0.175 & \dots & -0.859 & -0.595 \\ 3.747 & 1.115 & \dots & -0.418 & -0.084 \\ 1.383 & 1.184 & \dots & -0.493 & -0.562 \end{pmatrix}}^{S \text{ conditions}} \Bigg\} G \text{ genes}$$

(a) Gene expression matrix.



(b) Gene expression profiles.

Figure 3.1 ~ Gene expression data ~
(a) Gene expression matrix $\mathbf{M}$ *for 5 genes of* Saccharomyces cerevisae *(YNL325C, YLR014C, YDL243C, YJR106W and YGR145W) in 25 temporal conditions obtained with microarray experiments. (b) Representation of the corresponding gene expression profiles. Data are extracted from the mitotic cell cycle study of* S. cerevisae *in Spellman et al. (1998). Original time-course data are composed of 1631 genes and 25 temporal points.*

From these data, methods compute gene-gene interaction scores $\omega_{i,j}$ yielding a weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{G \times G}$ defining a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}; \omega)$, where $\mathcal{V}$ is the set of nodes — taking their indices in $\mathbb{V} = \{1, \dots, G\}$ — and $\mathcal{E}$ the set of edges. Genes can be split into two main categories: genes coding for transcription factors (TFs) (and metonymically denoted by TFs) and genes not identified to code for TFs (denoted by $\overline{\text{TFs}}$).

While the main majority of methods requires, after the computation of the adjacency matrix $\mathbf{W}$, an edge selection step to select a set of relevant edges $\mathcal{E}^*$ giving $\mathcal{G}^*$, there exist methods that directly provide the final GRN $\mathcal{G}^*$ by computing a sparse adjacency matrix — no additional thresholding step is thus required. A vast literature on GRN inference is available and we refer to Filkov (2005); Hecker *et al.* (2009); De Smet and Marchal (2010); Marbach *et al.* (2012); Emmert-Streib *et al.* (2012); Chai *et al.* (2014); Kurt *et al.* (2014) and Liu (2015) for meticulous reviews of the accessible approaches. We also refer to the R package *NetBenchmark* (Bellot *et al.*, 2015), an elegant tool to assess the robustness of around ten commonly cited GRN inference methods. Due to the profusion of GRN inference methods, establishing a well-separated typology of the

methods is difficult. However, it is usual to cleave GRN inference approaches into two classes of methods: metric-based or model-based. A third class, encompassing particular frameworks, can also be defined.

3.1.1 Metric-based inference

Metric-based methods involve the computation of a statistical measure reflecting the similarity or the dependence between pairwise — triplewise or more, in some cases — gene expression profiles. The two mostly used measures are related to correlation and mutual information.

~ *Correlation-based scores* ~ Integrating correlation-based methods in this overview can be discussed as they rather infer co-expression networks and they do not provide any causal interactions. Nevertheless, they can be complemental to other approaches and can provide some useful biological information and insights in terms of biological functionality (Stuart *et al.*, 2003). Several correlation-based measures, generically denoted by C, can be employed to construct the adjacency matrix $\mathbf{W}$ with elements $\omega_{i,j} = C(\mathbf{m}_i, \mathbf{m}_j)$. Among the mostly used correlation-based measures, we find the absolute or signed Pearson's and the Spearman's rank correlation coefficients. The Spearman's correlation is a Pearson's correlation computed on variable ranks, instead of variables themselves. Differences between these two measures reside in the kind of detected dependence: Pearson's correlation assesses linear relationships while Spearman's correlation assesses monotonic relationships. Weighted correlation network analysis (WGCNA) is a tool developed by Langfelder and Horvath (2008) to perform analysis from a gene correlation matrix. The proposed analysis encompasses module detection and validation, module relationships and key genes identification. Partial correlation can also be employed, but as it is generally estimated *via* Gaussian Graphical Models (GGM), we refer to Section 3.1.2 for a detailed description of partial-correlation-based methods. Nevertheless, the detected relationships using correlation metrics can be limited as rarely present in gene expression data. To overcome this limitation and extend the type of detected relationships, a large number of methods based on mutual information have been developed.

~ *Mutual information-based scores* ~ Mutual information is a measure quantifying the mutual dependence shared by stochastic phenomena. Let us define by X and Y, two random variables and their respective marginal probability $p(X = x)$ and $p(Y = y)$, simplified into $p(x)$ and $p(y)$ to lighten the notation. Given $p(x, y)$ the joint probability, the mutual information between two discrete random variables is defined as:

$$I(X,Y) = \sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right), \tag{3.1}$$

where $\mathcal{X}$ and $\mathcal{Y}$ are the set of discrete values taken by X and Y, respectively. In the case of continuous random variables, the summations over $\mathcal{X}$ and $\mathcal{Y}$ are replaced by integrals. However, mutual uinformtation in the continuous case can be estimated by finely discretizing the variables. Assimilating gene expression profiles $\mathbf{m}_i$ to random variables, the mutual information can thus be used to compute gene-gene interaction scores in the adjacency matrix $\mathbf{W}$.

This is strictly the case of the Relevance Network (RN) method proposed by Butte and Kohane (2000), where for each couple of genes $(i, j) \in \mathbb{V}^2$, elements in $\mathbf{W}$ are computed as $\omega_{i,j} = I(\mathbf{m}_i, \mathbf{m}_j)$. As gene expression data are generally modeled *via* continuous distribution, mutual information computation can require a discretization step. For this purpose, marginal probabilities are estimated by binning the data into a pre-defined number of discrete intervals and counting the number of data points within each bin. The same scheme is performed on the bi-variate histogram to estimate the joint probabilities. After mutual information computation, insignificant edges are removed by setting their weights to 0, leading to the final adjacency matrix $\mathbf{W}$. This step is based on a reference distribution of mutual information values, estimated by computing the mutual information on the randomized data. RN has been the first one to use mutual information for GRN inference context. Since then, a large number of methods have emerged or been extended.

Margolin *et al.* (2006) proposed the Algorithm for the Reconstruction of Accurate Cellular NEtwork (ARACNE), which firstly computes the matrix of mutual information using Gaussian Kernel estimators (Beirlant *et al.*, 1997) instead of the basic grid-based approach. Once mutual information is computed, insignificant weights are set to zero as in Butte and Kohane (2000). From the remaining non-null weights, an additional pruning step is performed, based on the Data Processing Inequality (DPI) property inherent to the mutual information. Indeed, for each existing gene-triplet, the lesser weighted of the three edges is removed. After these two corrective steps, the final adjacency matrix $\mathbf{W}$ is obtained.

The most adopted method using mutual information is called Context Likelihood of Relatedness (CLR) (Faith *et al.*, 2007). Initially, mutual information for each pair of genes is estimated *via* a B-spline smoothing and discretization of the data (Daub *et al.*, 2004). Then, to estimate the significance of the weights, a per gene null-distribution is used instead of a global null-distribution as in RN and ARACNE. For this purpose, for each pair of genes i and j, authors define p_i and p_j as the distribution of mutual information values computed for the gene i and j, respectively, against all genes $k \in \mathbb{V}$. Assuming a Gaussian distribution, a z-score can be computed for each of them. Based on these z_i- and z_j-scores, a joint likelihood measure is proposed: $\bar{z}_{i,j} = \sqrt{z_i^2 + z_j^2}$, defining the element of the adjacency matrix $\mathbf{W}$ i.e. in $\mathbf{W}$, the element $\omega_{i,j}$ is equal to $\bar{z}_{i,j}$.

The Minimum Redundancy NETworks (MRNET) proposed by Meyer *et al.* (2007) is based on the Maximum Relevance/Minimum Redundancy (MRMR) feature selection method (Ding and Peng, 2005). The use of this method in a GRN inference context is motivated by the fact that the MRMR criterion is an optimal pairwise approximation of the mutual information between two variables, conditioned by a set of selected variables. For each gene i, MRNET selects a subset of genes K — considered as potential partners — which maximizes a score s_i. This score s_i is defined as the difference of two terms. The fisrt one corresponds to the average mutual information between $\mathbf{m}_i$ and $\mathbf{m}_k$, for all $k \in K$. The second one is the average mutual information between $\mathbf{m}_k$ and $\mathbf{m}_{k'}$, for all $(k, k') \in K^2$. For a couple of genes i and j, we thus define the weight $\omega_{i,j}$ in the final adjacency matrix $\mathbf{W}$ as $\omega_{i,j} = \max\{s_i, s_j\}$.

The four exposed methods (RN, ARACNE, CLR and MRNET) related to mutual information are the most used in a GRN inference context and are gathered in the R package *minet* developed by Meyer *et al.* (2008). Although less used in practice, other methods based on mutual information exist.

Notably, we can mention the Conservative Causal Core (C3Net) method developed by Altay and Emmert-Streib (2010), which infers an undirected and unweighted network in two steps. Firstly, the mutual information value, for each couple of genes, is estimated using a parametric Gaussian estimator (Meyer *et al.*, 2007), and non significant weights are evaluated thanks to a re-sampling method as in RN or ARACNE. A second step is added: for each gene i, authors look for the gene j in a given neighbor $\mathcal{N}_i$ of i, that shares the maximal mutual information value and set the corresponding $\omega_{i,j}$ coefficient in $\mathbf{W}$ to 1. In the Mutual Information 3 (MI3) approach, Luo *et al.* (2008) pertinently assume that gene regulation may involve more than one TF. To take into account this hypothesis in the inference, for each gene i, they look for the couple of TFs (j, j') that maximize the three-way mutual information defined as the sum of two conditional mutual information values between the gene i and a TF given the other TF. Identifying such a couple of TFs leads to add two edges in the network by setting $\omega_{i,j} = \omega_{i,j'} = 1$. This procedure is repeated for each gene $i \in \mathbb{V}$ to assemble a final network. Edges forming cycles in the resulting network are finally removed. The Conditional Mutual Information (CMI) method, developed by Soranzo *et al.* (2007), is also based on a similar principle. They firstly estimate, for each gene triplet (i, j, k) the conditional mutual information $I(\mathbf{m}_i, \mathbf{m}_j|\mathbf{m}_k)$. Then, from an 1-valued adjacency matrix $\mathbf{W}$, they set weights $\omega_{i,j}$ to 0 if, after a thresholding, the conditional mutual information $I(\mathbf{m}_i, \mathbf{m}_j|\mathbf{m}_k) = 0$ for at least one gene k. Note that a combination of mutual information and conditional mutual information was proposed by Liang and Wang (2008) in the MI-CMI method to infer GRNs. Reshef *et al.* (2011) propose the Maximal Information Coefficient (MIC) measure of dependence. Let X and Y be two variables of dimension m and n, respectively. For each pairs (p, q), $p \in \{1, \ldots, m\}$ and $q \in \{1, \ldots, n\}$, authors compute mutual information values given by all the $p \times q$ quantification grids. The MIC corresponds to the highest normalized mutual information evaluated across all the considered grid. However, although this measure shows promising performance on various large biological datasets, its use on the too-often small gene expression datasets reaches limits and more complex estimators for mutual information have to be employed.

In addition to metric-based GRN inference methods, which are model-free, another facet of the literature deals with model-based approaches. We thus now give an overview of these methods.

3.1.2 Model-based inference

Gene regulatory networks can also be obtained *via* model-based methods including regression models, Gaussian graphical models, Bayesian graphical models, Boolean models or differential equations. We provide in this section some of the concepts behind these various approaches in the GRN context. Note that in this section, notations are model-dependent and do not refer,

for the majority, to the previously introduced notations.

~ *Regression models* ~ A gene regulatory network inference task can be viewed as a variable selection problem. Indeed, the GRN aims at discovering, for each gene, the set of its regulators. Commonly used variable selection approaches rely on — sparse — regression models (Hastie *et al.*, 2013, 2015; Chiquet, 2015). Let y_i be the expression level of a target gene in the i-th experimental condition and let the vector $\boldsymbol{y} \in \mathbb{R}^S$ gather expression levels of the target gene in the S experimental conditions. Similarly, let $x_{i,j}$ be the expression level of the potential gene predictor j in the condition i and let the matrix $\mathbf{X} \in \mathbb{R}^{S \times G}$ gather the gene expression levels of G potential predictors in S conditions. The linear model assumes that y_i, the gene expression level of the target gene in the i-th condition, can be written as the weighted sum of the gene expression levels of the potential predictors in the conditions i:

$$y_i \;=\; \beta_0 \;+\; \beta_1 x_{i,1} \;+\; \beta_2 x_{i,2} \;+\; \ldots + \beta_G x_{i,G} \;=\; \beta_0 \;+\; \sum_{j=1}^{G} x_{i,j}\beta_j \tag{3.2}$$

How to interpret this model? The underlying problem is to discover, among a set of potential predictors, the subset of predictors that is responsible for the observation of the gene target. Let us interpret the model for a given condition i. The observed gene expression level of the target gene y_i can be explained by a combination of gene expression levels of the potential predictors $\{x_{i,1}, \ldots, x_{i,G}\}$. The level of implication of the predictor j is encoded in the coefficient β_j. In other word, the coefficient β_j indicates the proportion of the activity of the predictor j which participates to the observed activity of the target. A coefficient β_j equal to 0 implies that the potential predictor j does not participate to a given gene activity and cannot be assimilated to a candidate TF for this target. Note that the coefficient β_0 is thus interpreted as the intercept of the regression.

Now, taking into account all the experimental conditions, (3.2) yields the compact form: $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta}$, where $\boldsymbol{\beta} = \{\beta_0, \beta_1, \ldots, \beta_G\}$ and $\mathbf{X} = \{\mathbf{1}, \mathbf{x_1}, \ldots, \mathbf{x}_G\}$ with $\mathbf{1}$ is one-valued vector of size S. The aim of the regression is to find the set of β_i values which minimize the difference between the observation $\boldsymbol{y}$ and the model $\mathbf{X}\boldsymbol{\beta}$. The ℓ_2 norm is commonly used to evaluate this discrepancy. In addition, regularized terms could be added to enforce particular behaviors of the coefficients to be estimated. Hence, the regression problem can thus be expressed as the following minimization problem (3.3):

$$\underset{\boldsymbol{\beta} \in \mathbb{R}^{G+1}}{\text{minimize}} \quad \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \;+\; \lambda\, \varphi(\boldsymbol{\beta}), \tag{3.3}$$

where $\varphi(\boldsymbol{\beta})$ encodes the regularization terms on β_i coefficients and λ denotes the regularization parameter. Note that optimization problem in (3.3) integrates the intercept β_0. Centering the data may avoid to include it in the optimization process (Hastie *et al.*, 2015). This generic regression model can be used to discover candidate TFs for each target gene and then construct a gene regulatory network. When $\lambda = 0$, the classical least squares problem is recovered.

However, without any constraint on the β_i coefficients, one could observe an excessive variance of the magnitude coefficients, leading to an unreliable prediction error. In order to control the variance, the regularization term could take the form of the squared ℓ_2 norm of the coefficients i.e. $\varphi(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|^2$. This class of ℓ_2 penalized regression problem is called Ridge regression and can be explicitly solved (Tibshirani, 1996). Note that it bears relations with Whittaker filters (Whittaker, 1922; Macaulay, 1931) alluded to in the section devoted to analytical data filtering. Unfortunately, this approach is rarely used in a GRN context and other penalties are preferred. Notably, instead of controlling the variance of the estimated parameters β, it can be judicious, in a variable selection strategy, to enforce sparsity in the coefficients. For this purpose, Tibshirani (1996) defines the regularization term as an ℓ_1 norm of the coefficients i.e. $\varphi(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$. This method, which has become extremely popular, is known as Least Absolute Shrinkage and Selection Operator (lasso). Enforcing a high number of null coefficients *via* the ℓ_1 penalty, lasso only selects a small number of candidate TFs, which is a coherent biological assumption and yields sparse networks. The two most popular algorithms existing to solve lasso are *active sets* (Osborne *et al.*, 2000) and LARS (Least Angle Regression and Selection) (Efron *et al.*, 2004). A large number of high-dimensional graphs (Meinshausen and Bühlmann, 2006) and specifically GRN inference methods rest upon lasso (van Someren *et al.*, 2005; Bonneau *et al.*, 2006; Meinshausen and Bühlmann, 2010; Haury *et al.*, 2012). Extensions to LASSO can be defined, as in the Bridge regression (Fu, 1998) or in the Elastic-net regression (Zou and Hastie, 2005), where the regularization term encompasses a sum of an ℓ_2 and an ℓ_1 norm i.e. $\varphi(\boldsymbol{\beta}) = \alpha\|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|^2$. The latter approaches can thus be viewed as a compromise between the Ridge and the lasso regressions and tend to select groups of correlated predictors. The GRN inference method proposed by Shimamura *et al.* (2010) is inspired from the Elastic-net regression. In the same vein as Elastic-net, the Group-lasso approach, developed by Yuan and Lin (2006), enforces all coefficients in a group of correlated predictors to become nonzero (or zero) simultaneously. Such grouping strategies inspired Liu *et al.* (2014) for GRN inference. A sparse version of the Group-lasso was designed by Simon *et al.* (2013) to promote sparsity either in groups and within each group. In a slightly different application, authors in Obozinski *et al.* (2011) demonstrates the interest of such group lasso strategy for cancer prediction from gene expression data. In the Cooperative-lasso approach, Chiquet *et al.* (2012) propose to promote sign coherence and variable selection within each group by modifying the Group-lasso penalty. The Fused-lasso, introduced by Tibshirani *et al.* (2005) was developed to deal with time-series data. The regularization term encompasses a sum of ℓ_1 norms, one acting on the coefficients and another acting on the difference between two adjacent coefficients. While the first penalty enforces sparsity in the coefficients — as in the lasso — the second one enforces sparsity in their differences, allowing us to drive coefficients to vary in a smooth manner. This assumption effectively makes sense in time varying gene expression data leading to satisfying GRNs (Omranian *et al.*, 2016). A Weighted-lasso strategy was designed in Charbonnier *et al.* (2010) to deal with time-series data and to take into account the underlying time structure. Finally, in the bLARS approach developed by Singh and Vidyasagar (2016), the authors make the judicious assumption that the expression level of a target gene could be expressed as a weighted linear sum of potentially non-linear functions of the expression levels of the predictors.

Whatever the opted regression strategy used to infer a GRN, choosing appropriate regular-

ization parameters could be challenging. We recall that these parameters play an important role as they control the influence of the penalties on the global regression. Bootstrapping (Efron, 1979) and cross-validation (Efron, 1983) offer suitable re-sampling strategies to select relevant regularization parameters for a regression model. Note that the cross-validation is preferred for high-dimension data. While regression-based methods can lead to satisfying results on *in-silico* dataset, they can falter on real data Marbach *et al.* (2012), even with optimal regularization parameters. This downturn could be explained by the scarcity of the number of experimental conditions with respect to the number of genes. Indeed, in such a case, the regression problem becomes highly undetermined and generates less accurate GRNs. Other limitations of using regression-based methods for GRN inference purpose can be recovered in Gadaleta (2015).

We now present another model-based approach relying on probabilistic graphical models. In a GRN inference context, they can be decoupled into two main frameworks: Gaussian Graphical Models and Bayesian networks.

~ *Probabilistic graphical models* ~ A Probabilistic Graphical Model (PGM) is a probabilistic model representing random variables and their dependencies *via* a graph structure. In such a graph, nodes corresponds to random variables. The presence of an edge $e_{i,j}$ between nodes v_i and v_j encodes a dependence between random variables X_i and X_j, conditionally to the other random variables. Conversely, the absence of an edge between nodes v_i and v_j reflects a conditional independence between random variables X_i and X_j. Assuming that the gene expression data — more precisely gene expression profiles — are random variables, graphical models can model gene regulatory networks (Friedman *et al.*, 2000). The key challenge of the GRN inference from PGM framework is to compute all the conditional dependencies between random variables. Various strategies are employed following the assumptions made on the random variable distributions.

Let us define by $X = (X_1, \dots, X_G)^\top$ a random vector, where, for all $i \in \{1, \dots, G\}$, the random variable $X_i = (x_1, \dots, x_S)$ corresponds to the gene expression profile of the gene i over the S experimental conditions. If the random vector X follows a multivariate Gaussian distribution the underlying PGM belongs to Gaussian Graphical Models (GGM) (Whittaker, 1990). As frequently used in a GRN inference context, we firstly give an overview of GGM-based methods to infer GRN, before extending this overview to more general PGM.

Why GGM seem convenient for GRN inference? In the GGM, random vector X is assumed to be multivariate Gaussian with a distribution parametrized by a zero-mean and a dispersion or covariance matrix $\mathbf{\Sigma} = (\Sigma_{i,j})_{(i,j)\in\mathbb{V}^2}$. The inverse of the covariance matrix, denoted by $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$, is classically named as the *precision* (or *concentration*) matrix. Assuming $\mathbb{V}\backslash(i,j)$ be the set of all indices taken off the couple (i,j), the element $\Omega_{i,j} = \operatorname{cov}\left(X_i, X_j \,|\, X_{\mathbb{V}\backslash(i,j)}\right)$ encodes the dependence between random variables X_i and X_j, conditional on all other variables $X_{\mathbb{V}\backslash(i,j)}$. Moreover, from the conditional dependencies in $\mathbf{\Omega}$, the partial correlation $\rho_{i,j}$ between random variables X_i and X_j can be recovered thanks to the following scaling relation: $\rho_{i,j} = -\frac{\Omega_{i,j}}{\sqrt{\Omega_{i,i}\Omega_{j,j}}}$ (Dempster, 1972). In a GRN context, the partial correlation plays an important role by remov-

ing indirect edges. As mentioned in Section 2.4, this kind of edges is one of the main sources of false positive edges. Hence, for the gene triplet $(i, j, k) \in \mathbb{V}^3$, if the following regulation scheme exists: $i \to j \to k$, the correlation between X_i and X_k could be give a non-null value (yielding an edge between nodes v_i and v_k) while the partial correlation will be null (absence of an edge between nodes v_i and v_j). Finally, reconstructing a GGM is equivalent to estimating the precision matrix $\mathbf{\Omega}$ (Lauritzen, 1996). In our context, the rescaling of $\mathbf{\Omega}$ could directly yield the adjacency matrix of the GRN.

Dealing with GGM chiefly rests upon the estimation of the precision matrix $\mathbf{\Omega}$. We propose here to only highlight the main approaches in a GRN inference context and refer to the work of Fan *et al.* (2016) for a more complete overview regarding the concentration matrix estimation. In Statistical Inference for Modular Networks (SIMoNe), developed by Ambroise *et al.* (2009), a regularized likelihood criterion, involving a latent structure on the expected network, is defined. The chosen ℓ_1 penalty on the latent structure enforces a sparse network. An Expectation-Maximization (EM) algorithm, embedding a lasso-like procedure, is then employed to estimate the precision matrix according to the designed criterion. Meinshausen and Bühlmann (2006) also deal with a lasso-like procedure in order to estimate the concentration matrix $\mathbf{\Omega}$. Their penalty, promoting sparse network, is based on a neighborhood selection approach consisting in finding, for each variable i, a subset of variables, denoted by $\mathcal{N}_i$, such that the random variable X_i is conditionally independent of all the remaining random variables X_k, $k \notin \mathcal{N}_i$. The R package *geneTS*, developed by Schäfer and Strimmer (2005), reconstructs a GGM *via* a statistical framework embedding a shrinkage estimator. In the same vein, Li and Gui (2005) want to enforce the sparsity by defining a cost function depending on the off-diagonal elements. They used a Threshold Gradient Descent (TGD) regularization algorithm to solve the problem and estimate a sparse network. Unlike approaches focused on a parsimonious *a priori*, Wille *et al.* (2004) propose to construct a GGM for each gene-triplets. Hence, they determine all dependencies between two genes, conditionally to a third one, before to aggregate the generated sub-networks into the final network specifying the GRN. Linear dependencies resulting from similar gene expression profiles generally pollute the precision matrix. Toh and Horimoto (2002) limit their presence by constructing a GGM on the averaged gene expression profiles obtained *via* a clustering approach. The resulting graph is not — *strictly speaking* — a GRN as it encodes the conditional dependencies between clusters of genes instead of genes themselves.

Although GGM have been largely used to infer GRN, their restriction to linear dependencies may generate inaccurate graphs in practice. Indeed, linear dependencies do not reflect combinatorial regulations i.e. when TFs have to act in synergy to regulate another gene, see Figure 4.2(a) - p. 81. We now present a brief review of the more general probabilistic graphical models developed for GRN inference task: the Bayesian network (BN).

A Bayesian network is defined as a directed and acyclic graph (DAG) $\mathcal{G}$ with a set of nodes $\mathcal{V}$ corresponding to random variables $X_1, \ldots, X_G$. Conditional (or local) probability distributions per variable, parametrized by θ, allow us to determine the structure of the graph. The resulting graph is a representation of a joint probability distribution (Friedman *et al.*, 2000). Hence, Bayesian inference aims at finding, among the set of possible graphs parametrized by θ,

the graph structure that fits at best the data (given by the random variables). This inference is performed by generating all possible graphs, scoring them and keeping the best-scoring network. Let us now give the general framework used to define an appropriate Bayesian score. In the following, we permit ourselves to get nodes and variables X_i mixed up.

Let us first introduce some specific vocabulary. Given a directed edge between variables X_i and X_j i.e. $X_i \to X_j$, X_i refers to a parent of X_j, while, conversely, X_j referred to as a child, or a descendant, of X_i. The main assumption involved in the Bayesian network framework rests upon the Markov assumption: *given its parents, each node is independent of its non-descendants.* The joint probability distribution of the graph can thus be expressed as the product of local probability distributions:

$$P(X_1, \ldots, X_G) = \prod_{i=1}^{G} P(X_i \,|\, pa(X_i)), \tag{3.4}$$

where $pa(X_i)$ denotes the set of parents of X_i. We refer to Figure 3.2 for a toy example of a Bayesian network $\mathcal{G}$ and the associated joint probability distribution.

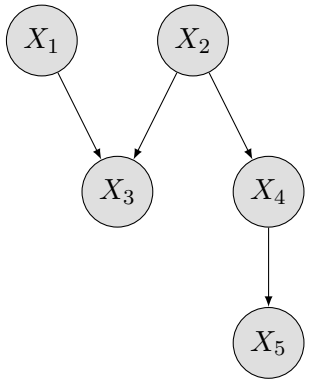


Figure 3.2 ~ A Bayesian network ~
This Bayesian network is composed of five nodes. Only three of them have parents: X_3, X_4 and X_5. The local probability for parent-free nodes X_1 and X_2 are $P(X_1)$ and $P(X_2)$, respectively. Local probability distribution of X_3 is $P(X_3 \,|\, X_1, X_2)$, while the ones of X_4 and X_5 are $P(X_4 \,|\, X_2)$ and $P(X_5 \,|\, X_4)$, respectively. Hence, the associated joint probability distribution is given by: $P(X_1, X_2, X_3, X_4, X_5) = P(X_1)\, P(X_2)\, P(X_3 \,|\, X_1, X_2)\, P(X_4 \,|\, X_2)\, P(X_5 \,|\, X_4)$.

As previously mentioned, a Bayesian score has to be defined in order to select the best network in terms of data fitting (Heckerman *et al.*, 1995). This score is defined as the posterior probability of a graph given the data: $s(\mathcal{G} : D) = \log P(\mathcal{G} \,|\, D)$ (Friedman *et al.*, 2000). Using Bayes' rule, this score can be re-expressed as:

$$s(\mathcal{G} : D) = \log(P(D \,|\, \mathcal{G})) + \log(P(\mathcal{G})) + C,$$

where C is a negligible constant and $P(D \,|\, \mathcal{G})$, which is the marginal likelihood, reflects the average probability of the data over all possible parameters θ assigned to G. According to the *prior* chosen for the conditional probabilities, an adapted algorithm has to be designed. In view of the exhaustive variety of BN-based approaches developed for GRN inference, we refer to Pe'er *et al.* (2001); Tamada *et al.* (2003); Werhli and Husmeier (2007); Vignes *et al.* (2011) and Young *et al.* (2014) for some examples. On a similar principle, dynamic BN was developed to tackle time-series data and to discover the dependencies that exist between genes in a temporal process (Perrin *et al.*, 2003; Yu *et al.*, 2004; Dojer *et al.*, 2006; Vinh *et al.*, 2012).

Although BN-based approaches inspire the community working on GRN inference, their utility in practice is limited to small networks, often characterized by a number of genes (variables) having the same order of magnitude than the number of experimental conditions (observations). This is due to the fact that, even with a reduction of the space of possible solutions, a large number of possible networks has to be generated to find the best one. Unfortunately, an overwhelming majority of real data gathers a large number of genes (more than thousands of genes), for which a low number of experimental conditions is available. Hence, BN may suffer from this high number of variables in addition to a lack of balance between variables and observations.

In the two following sections, we introduce GRN inference methods specially well-adapted to time-series data. They encompass Boolean models and differential equations models.

~ *Boolean-network-based models* ~ Before presenting the methodology to infer a GRN *via* Boolean models, let us recall some basics on the Boolean logic. A Boolean variable x can take two logical values only: *true* or *false*, usually denoted by 1 or 0. Three logical operators are used to deal with Boolean variables: **and**, **or** and **not**. Table 3.1 summarizes the rules for each of them.

Input		Output
x	y	x **and** y
0	0	0
0	1	0
1	0	0
1	1	1

(a) Rules for **and** *operator.*

Input		Output
x	y	x **or** y
0	0	0
0	1	1
1	0	1
1	1	1

(b) Rules for **or** *operator.*

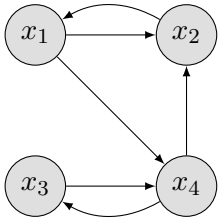
Input	Output
x	**not** x
0	1
1	0

(c) Rules for **not** *operator.*

Table 3.1 **~** TRUTH TABLES FOR LOGICAL OPERATORS **and**, **or** AND **not** **~**
Variables x and y refers to Boolean variables valued by 0 or 1. Truth tables summarizing logical operator rules are given for operators **and** *(a),* **or** *(b) and* **not** *(c).*

A Boolean function f is a function of Boolean variables connected by logical operators: $f(x_1, x_2, x_3) =$ **not**$(x_2$ **and** $(x_1$ **or** $x_3))$, for instance. A Boolean network (BoN) is a directed graph where nodes correspond to Boolean variables. At each node x_i is associated a Boolean function f_i, depending on the parent nodes of x_i only. Hence, Boolean functions encode network topology, see Figure 3.3. Boolean networks were firstly established in a biological context by Kauffman (1969). An important notion in Boolean networks is the *state* of the network which encodes the node values at a given time. It is defined, at each time and for the whole network, as $S(t) = (x_1(t), \ldots, x_G(t))$. From two consecutive times, node values in $S(t)$ are updated thanks to the Boolean functions to give the new state $S(t+1)$. The update is simultaneously performed for each node i by $x_i(t+1) = f_i(x_{i,1}(t), \ldots, x_{i,P}(t))$, where P is the number of parent nodes of x_i. The $S(t)$ to $S(t+1)$ computation is called the *state transition*.

On Boolean networks and GRN... A BoN deals with binary-valued nodes. In the general GRN



$$f_1(x_2) = x_2$$
$$f_2(x_1, x_4) = x_1 \textbf{ or } x_4$$
$$f_3(x_4) = \textbf{not } x_4$$
$$f_4(x_1, x_3) = x_1 \textbf{ and } x_3$$

Figure 3.3 ~ NETWORK TOPOLOGY AND UNDERLYING BOOLEAN FUNCTIONS ~
Boolean network (BoN) composed of five nodes. As node x_1 has one parent node (x_2), its associated Boolean function f_1 only depends on the node x_2. The same scheme is observed for x_3 which only has a unique parent x_4. Node x_2 has two parents x_1 and x_4, also corresponding to the variables of the Boolean function f_2. Similar concept is applied for node x_4. Logical operators involved in functions do not act on the topology but on the node value only.

context, these nodes correspond to genes and their values to gene expression levels. Assimilating a BoN to a GRN requires a discretization of the gene expression data into two levels. Hence, at each time, node values are known and correspond to the activation (1 valued) or the non-activation (0 valued) of genes. In this case, all network states are known — one state corresponding to one experimental condition. The unknowns are the functions allowing the transition from a state to another. These functions have to be determined to fit the data i.e. to obtain the known gene activation status given by the data. Once Boolean functions are determined, the GRN is spontaneously built. Indeed, we recall that Boolean functions directly provide the network topology by encoded relation between nodes. In addition to the topology, BoN are useful for biological interpretation as for each node, a Boolean function encodes the regulation effect of each of its parent nodes assimilated to TFs. In addition to an easy interpretation, the dynamical properties of Boolean networks favor their uses to model GRN (Kaderali and Radde, 2008; Wang *et al.*, 2012b).

As mentioned, the key challenge is to determine the correct Boolean functions, in terms of data fitting. For this purpose, several approaches have been developed. Fixing the number of parent nodes to k, Akutsu *et al.* (1999) find a GRN consistent with the data by trying out all Boolean functions of k variables among G. The REVEAL approach, developed by Liang *et al.* (1998), integrates mutual information computation between consecutive states to reduce the space of possible solutions. Ideker *et al.* (2000) also exploit information-theoretic measure to determine consistent Boolean functions from a set of identified parent nodes. A decision tree inference algorithm, mimicking a Boolean network, is used in Silvescu and Honavar (2001) to infer a GRN from time-series data. We refer to Saadatpour and Albert (2013) for a more exhaustive overview. Note that the synchronous assumption used to update states is poorly realistic. To overcome this drawback, probabilistic Boolean networks can be employed as in (Shmulevich *et al.*, 2002; Pal *et al.*, 2004), for instance.

Although Boolean networks provide a dynamical modeling of a GRN, the data discretization

into two levels only can be prejudicial for the inference. Another model-based approach, to dynamically infer GRNs, relies on differential equations, sometimes coupled with one of the previously presented frameworks.

~ *Differential equations models* ~ Differential equations are used to model the rate of change of gene expression as a function of the expressions of other genes. Such kind of modeling allows us to determine, for a pair of genes, whether an interaction exists, which is the regulator, the effect (activation or repression) and the strength of the regulation. The identification of these dynamical and causal relationships allows the construction of the GRN. We focus this brief review on Ordinary Differential Equations (ODE) and refer to de Jong (2002) for more details on the potential use of Partial Differential Equations (PDE).

Formally, let $x_i(t)$ be the expression level of the gene i at the time t, the rate of change of the expression of gene i can be expressed, in its generic form, as:

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = f_i(x_1(t), \ldots, x_G(t), p), \tag{3.5}$$

where p is the set of parameters of the system and f_i is a function describing the rate of change. This function f_i combines expression levels of all genes to produce the rate of change of the gene i. Note that in some cases, the function f_i can depend on a restricted number of genes only, corresponding for instance to TFs. An elementary classification of ODE-based approaches relies on the type of functions f: linear or non-linear (Hecker *et al.*, 2009). Although non-linear functions are more realistic to describe gene regulatory mechanisms, linearized additive models as in (3.6) are the most employed:

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = \beta_{i,0} + \beta_{i,1}x_1(t) + \ldots + \beta_{i,G}x_G(t), \tag{3.6}$$

where $\beta_{i,j}$ are coefficients to be determined. Coefficient $\beta_{i,j}$ reflects the strength of the regulatory effect of the gene j on the gene i. From (3.6), defining the whole system for each gene, is then possible. The resulting system can be viewed as a regression problem, where the optimal $\beta_{i,j}$s coefficients directly provide the elements of an adjacency matrix encoding the GRN. As mentioned in Section 3.1.2, additional constraints can be added to reduce the state of possible solutions. Hence, adding a sparsity constraint on the network can be modeled by an ℓ_1 norm on $\beta_{i,j}$s coefficients. In such a case, a lasso-like problem is recovered and we refer to Section 3.1.2 for its resolution. In Yeung *et al.* (2002), authors propose to evaluate the space of possible solutions through a Singular Value Decomposition (SVD) procedure and then perform an ℓ_1-based regression method to chose the sparsest one. A similar approach was developed in Wang *et al.* (2006). It was designed to take into account several datasets and provide a consensus sparse network. Other approaches integrating a sparsity assumption were developed in Weaver *et al.* (1999) or Chen *et al.* (1999), for instance. Lu *et al.* (2011) propose to firstly perform a gene clustering. Then, instead of dealing with genes, the authors use mean expression curves, given by averaging gene expression profiles in the same cluster, to construct the linear ODEs. Hence, they try to evaluate, for a given cluster, the regulatory effect of the other gene clusters. Their linear ODEs are defined to encode sparsity through a Smoothly Clipped Absolute Deviation (SCAD) penalty

(a lasso-like penalty allowing variable selection). The resulting pseudo-regression problems are solved *via* the CCCP-SCAD algorithm (Kim *et al.*, 2008). This ODE-based inference is embedded in the tool D-NetWeaver (Wu *et al.*, 2014). In addition to network inference, this tool includes some classical gene expression data processing such as the identification of differentially expressed genes, functional enrichment analysis and gene clustering. A different approach, NARROMI, developed by Zhang *et al.* (2013), combines linear ODEs and information theoretic metrics to infer a reliable network by eliminating indirect regulations. We refer to the work of Bansal *et al.* (2007) and Polynikis *et al.* (2009) for a comparative study of differential-equation-based approaches to infer GRN.

Alongside metric-based and model-based methods, other methods have also been developed, for which we now give a brief overview.

3.1.3 Ancillary inference methods

Miscellaneous GRN inference methods cover neural networks, supervised learning or statistical analysis. For each of these frameworks, we provide an overview of the methodology employed to determine a GRN, through particular examples from the literature.

Küffner *et al.* (2012) assume that relevant relationships between a TF and its target genes lies on mutual dependence of their expression in a subset of experimental conditions, at least. As introduced in Section 3.1.1, dependence has been largely evaluated through correlation or mutual information. However, correlation measures are restricted to linear dependencies and mutual information requires a discretization of the data. To overcome these two drawbacks, Küffner *et al.* (2012) propose to determine dependence *via* a non-parametric and non-linear correlation coefficient η^2 — derived from an analysis of variance with ANOVA — without data discretization. Their ANOVA models: *i*) effects of the differential expression across the experimental conditions, *ii*) whether the gene expression profiles differ and *iii*) the joint effects of the two formers. Hence, they can evaluate each of them through a sum of squares decoupling strategy. The sum of squares quantities reflect dispersion measures, allowing us to define their correlation coefficient η^2 as the fraction of the total variation that is explained by the differential gene expression across experimental conditions.

Supervised-learning-based methods induce a wind of change in the construction of the GRN. The best example is GENIE3, developed by Huynh-Thu *et al.* (2010), an elegant tree-based approach actually belonging to the top-performing GRN inference methods. The main postulate in GENIE3 relies on the fact that the expression level of a gene i in a given condition j, denoted by $x_{i,j}$, is a function of the complementary set, denoted by $x_{-i,j}$ and corresponding to the other gene expression levels in the same condition:

$$x_{i,j} = f_i(x_{i,j}, \ldots, x_{i-1,j}, x_{i+1,j}, \ldots, x_{G,j}) = f_i(x_{-i,j}). \tag{3.7}$$

Based on this statement, GENIE3 treats the GRN inference problem as multiple feature selection problems. For each gene i, they aim at finding, among the set of variables $x_{-i,j}$, how they explain the observation $x_{i,j}$. Traditionally, the feature selection problem returns a subset of explicative

variables. Authors of GENIE3 prefer a ranking of the whole set of variables. As one feature selection problem is computed for each gene, a local ranking is assigned to each gene. Then, these local rankings are combined to a global one to yield the GRN. For each sub-problem i, concerning the gene i, the learning of the function f_i can thus be obtained by minimizing (3.8):

$$\sum_{j=1}^{S} \left(x_{i,j} - f_i(x_{-i,j})\right)^2, \tag{3.8}$$

where we recall that S is the number of experimental conditions. This problem can be solved *via* regression trees (Breiman *et al.*, 1984; Izenman, 2008).

Barking up regression trees. Regression trees are also called prediction trees. These particular graphs are built through a recursive process consisting in a binary partitioning of predictive variables. Two kinds of nodes are involved: interior nodes encoding a test on the predictive variables and terminal nodes encoding the predicted value for the output, see Figure 3.4. At each level of the tree, the best split is found by minimizing the empirical variance of the output variable in the generated partition. This optimization part leads to the definition of the test to be applied on predictive variables. Then, the algorithm reiterates the splitting on each branch, using the former rule, until a stopping criterion. This criterion can be a minimum node size (partition with a minimum number of variables) or when a terminal node is reached. Once the regression tree is built, the computation of a variable importance measure can be performed, leading to a ranking of the predictors with respect to the output prediction.

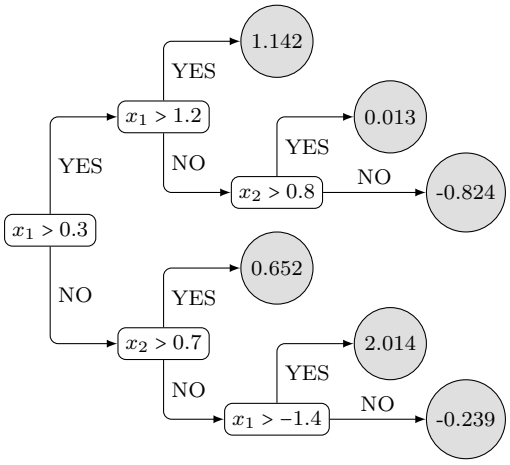


Figure 3.4 ~ EXAMPLE OF A REGRESSION TREE ~
The toy example involves 2 predictive variables, x_1 and x_2, and an output variable y. White nodes correspond to test on predictive variables while gray nodes corresponds to terminal nodes encoding the value predicted by the subset of corresponding predictive variables. For instance, the prediction value 1.142 is obtained for variable $x_1 > 1.2$, whatever the variable x_2.

In a first instance, as many regression trees as genes have to be constructed. Nevertheless, ensemble methods highly improve single tree construction. An ensemble method consists in

constructing various trees, with underlying randomization, and in averaging predictions for the various trees. Random Forests (Breiman, 2001) and Extra-Trees (Geurts *et al.*, 2006) are the two ensemble methods chosen by the authors of GENIE3. For each sub-problem i, ensemble trees predictions correspond to genes ranking i.e. for the regression tree related to the gene i, $G-1$ weights $\omega_{i,j}$ are returned, with $j \in \{1,\dots,i-1,i+1,\dots,G\}$. These weights can thus be directly interpreted as elements of the adjacency matrix of the inferred GRN. Note that, using this procedure, GENIE3 provide non-symmetric weights allowing us to generate a directed GRN. In the same vein, other tree-based approaches exist to infer GRN (Soinov *et al.*, 2003; Haury *et al.*, 2012; Ruyssinck *et al.*, 2014; Huynh-Thu and Sanguinetti, 2015) or identify regulatory programs (Segal *et al.*, 2003; Joshi *et al.*, 2009). Unlike previous tree-based approach, SIRENE, developed by Mordelet and Vert (2008), used a Singular Value Decomposition (SVD) procedure to identify, for each TF, its gene targets. In addition to the gene expression data, the method requires a list of known interaction between TFs and their targets, as well as when available, a list a negative interactions to learn the classifier.

Neural networks also address GRN inference. Neural networks, as the name suggests, take inspiration from animal's nervous system and mimic the synapse/neuron connection functioning. The activity of a neuron j is driven by its connection with numerous synapses. Each synapse i is defined by its state x_i (the entry) and interacts with the neuron j *via* a synaptic coefficient $\omega_{i,j}$. The action potential of the neuron j is defined as the weighted sum of the entries. An activation function g is then applied on the action potential to determine whether the neuron j is activated, with respect to the information coming from its synapses. Classically, thresholding functions — or their soften versions: sigmoid functions, for instance — are considered for the activation function g. Figure 3.5 illustrates this functioning.

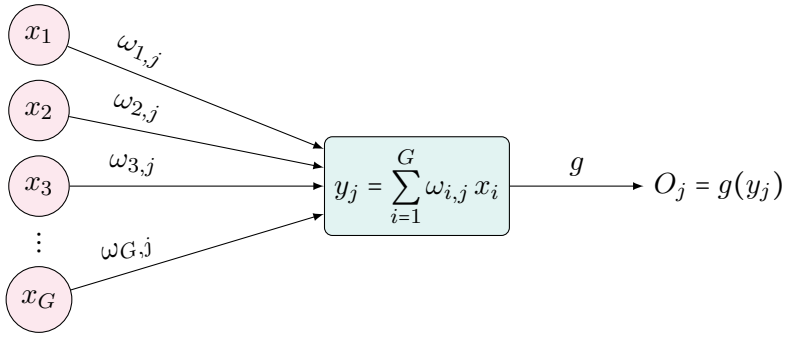


Figure 3.5 ~ Synapse/neuron connection functioning ~
The neuron j, rectangle node in green, receives information from G synapses (pink nodes), characterized by their states x_i, $i \in \{1,\dots,G\}$. Each synapse i acts on the neuron j with a strength $\omega_{i,j}$. The action potential, corresponding to the weighted sum of the entries, is denoted by y_j. Then, the output state of the neuron j, denoted by O_j, is driven by the function g applied on the action potential.

From this apparently simple process, an elegant, first-order analogy with gene regulation can be made. A gene to be regulated is assimilated to the neuron while its potential regulators

are assimilated to the synapses. The state of the potential regulator x_i corresponds to the gene expression level of the corresponding gene. Weight $\omega_{i,j}$ linking a potential TF i to a gene target j reflects the strength of the action of the potential regulator on its target. The function g encodes the global regulatory effect of the combined regulators. However, continuous-time recurrent neural networks are preferred in order to refine the gene regulatory mechanisms. They are able to model either nonlinear or dynamic interactions among genes thanks to ordinary differential equations (Ressom *et al.*, 2006). In GRN context, such recurrent neural networks can model $\dot{x}_j$, the rate of change of gene expression j, by:

$$\tau_j \dot{x}_j = g\left(\sum_{i=1}^{G} \omega_{i,j}\, x_i \;+\; \beta_j\right) - \lambda_j\, x_j\,, \tag{3.9}$$

where τ_j, β_j and λ_j refer to a time constant rate, the basal expression level and the reaction decay rate of the gene target j, respectively. In this model, only weights $\omega_{i,j}$ are unknown and have to be determined. This parameter estimation is performed through a scoring function to optimized. This scoring function corresponds to either network performance or error measure, for instance. This framework was used in Wahde and Hertz (2000); Blasi *et al.* (2005) and it was adapted by Xu *et al.* (2007) to integrate external variables into the model. These external variables reflect added exogenous inputs such as chemicals or nutriments, for instance. In Lee and Yang (2008), authors firstly perform a gene clustering *via* a self-organizing (feature) map for (SOM/SOFM) procedure (Kohonen, 2000). Hence, they obtain smaller sets of genes, in which recurrent neural networks are used. The advantage of such an approach is to construct smaller networks with the same number of data. Indeed, the ratio between the number of genes and experiments is decreased. We can thus expect increased accuracy of the inferred global network. A different neural network model is used in Günther *et al.* (2009), where the authors focused on a feed-forward multilayer perceptron model. Naturally, the recent inception of the deep learning paradigm yielded incursions into bioinformatics (Min *et al.*, 2016) and gene network inference (Chen *et al.*, 2016).

The proposed review of GRN inference methods is by no means exhaustive. Due to the profusion of literature in this field, we chose to only focus on the main approaches and related methods. We now introduce an important aspect in the development of GRN inference methods: their validation.

3.2 Evaluation methodology

In this section, we give some details about the different datasets used to validate our developed approaches and state-of-the-art methods used to compare them. Objective performance metrics for network inference are then discussed as well as methodology for biological interpretation of inferred networks. A similar review is given for clustering purposes.

3.2.1 Datasets and methods

In order to rigorously validate our developed GRN inference methods — *BRANE Cut* (Pirayre *et al.*, 2015a), *BRANE Relax* (Pirayre *et al.*, 2015b) and *BRANE Clust* (Pirayre *et al.*, 2018a) — we

used datasets provided by the challenges DREAM4 and DREAM5. Once the validation is acted on simulated and real benchmark datasets, application to in-house *Trichoderma reesei* data can be considered.

~ *DREAM4 and DREAM5 datasets* ~ The DREAM4 multifactorial challenge is composed of five datasets of simulated gene expression data. We recall that the simulation protocol is given in Section 2.2.3. Each dataset is composed of 100 genes simulated in 100 conditions that mimic multifactorial perturbations. In the challenge setting, information regarding what gene is a transcription factor is not given. For each dataset, the underlying *in silico* network is provided as ground truth. Source networks used to construct the *in silico* networks correspond to those of *Escherichia coli* (*E. coli*) and *Saccharomyces cerevisae* (*S. cerevisae*). The *E. coli* source network (Gama-Castro *et al.*, 2008) is composed of 1502 nodes and 3587 edges while *S. cerevisae* (Balaji *et al.*, 2006) compiles 4441 nodes and 12873 edges. Thanks to *in silico* networks, a list of transcription factors can be extracted to overcome the lack of information on them. Table 3.2 summarizes essential characteristics regarding the five datasets of DREAM4 and associated reference networks.

Network	1	2	3	4	5
S	100	100	100	100	100
G	100	100	100	100	100
# TFs	41	36	44	41	34
# TEs	176	249	195	211	193

Table 3.2 **~** Characteristics of DREAM4 multifactorial datasets **~**
*Number of experimental samples S, genes G and transcription factors (*TFs*) for the five datasets of the DREAM4 multifactorial challenge. We also report the number of true edges in the gold standard (# TEs).*

However, despite the efforts to simulate realistic data, DREAM4 datasets do not exactly reflect real datasets in two main underdeterminacy aspects (Siegenthaler and Gunawan, 2014): the ratio between the number of genes and the number of TFs on the one side, and the ratio between the number of genes and number of conditions on the other side. Indeed, in real data, the proportion of TFs is generally less than 10 % while the number of conditions is much lower than the number of genes. We note here that this latter dimensionality characteristic essentially causes difficulties to infer reliable networks as a few number of observations (conditions) is available for a large number of variables (genes). The two exposed deviations from reality may be prejudicial in a rigorous evaluation context. To overcome these defects, DREAM5 is composed of more realistic simulated data in addition to real data.

The challenge DREAM5 contains one simulated dataset and three real compendium datasets of *Staphylococcus aureus* (*S. aureus*), *E. coli* and *S. cerevisae*, respectively. For the first dataset, the ground truth corresponds to the *in silico* network used to generate simulated gene expression

data. Ground truth for *E. coli* and *S. cerevisae* compendia are obtained using RegulonDB (Gama-Castro *et al.*, 2011) (a reference database offering curated knowledge of the regulatory network and operon organization) and the study of MacIsaac *et al.* (2006), respectively. As the ground truth for *S. aureus* is uncertain in addiation to be poorly informative, no validation is performed on this dataset. Table 3.3 provides major characteristics for DREAM5 datasets and associated ground truths. Note that working on various model micro-organisms could be beneficial for the validation part.

Network	1 (*in silico*)	2 (*S. aureus*)	3 (*E. coli*)	4 (*S. cerevisae*)
S	805	160	805	536
G	1643	2810	4511	5950
# TFs	195	99	334	333
# TEs	4012	518	2066	3940

Table 3.3 ~ CHARACTERISTICS OF DREAM5 DATASETS ~
*Number of experimental samples S, genes G and transcription factors (*TFs*) for the four datasets of the DREAM5 challenge. We also report the number of (true) edges in the gold standard (# TEs).*

Alternatively to the *E. coli* dataset provided on the DREAM5 challenge, another dataset from *E. coli* — firstly introduced in Faith *et al.* (2007) — can also be used. It is composed of 4345 gene expression profiles, each profile containing 445 gene expression levels. This compendium contains both *steady-state* and *time-course* expression profiles. As in Faith *et al.* (2007), we used the RegulonDB 3.9 to evaluate inferred networks. This database offers a set of 1211 genes for which 3216 regulatory interactions are confirmed.

~ Proprietary real data on Trichoderma reesei ~ The filamentous fungus *Trichoderma reesei* is used at IFPEN for its capability to produce cellulases, enzymes used in the second generation bio-fuel production process to convert cellulose contained in plant to simple sugar such as glucose. For several decades, various strains of *T. reesei* were generated from the wild-type strain `Qm6a` by random mutagenesis. Among the generated strains, two hyper-producer strains are selected: `NG14` and `Rut-C30`, where `Rut-C30` exhibits a higher productivity than `NG14`.

In order to better understand the cellulase production of these hyper-producer strains, IFPEN researchers studied the whole gene expression of `Rut-C30` and `NG14` on conditions favorable to cellulase production such as on lactose culture medium. Complementing transcriptomic data with genome organization data, this previous study (Poggi-Parodi *et al.*, 2014), surprisingly shows an essentially intact induction system in cellulase hyper-producer *T. reesei* strains. Moreover, in the study of Jourdier *et al.* (2013), authors highlight differential enzyme activities according to the proportion of inducer, in industrial conditions. Based on these statements, additional internal experiments are designed to produce novel data in order to refine knowl-

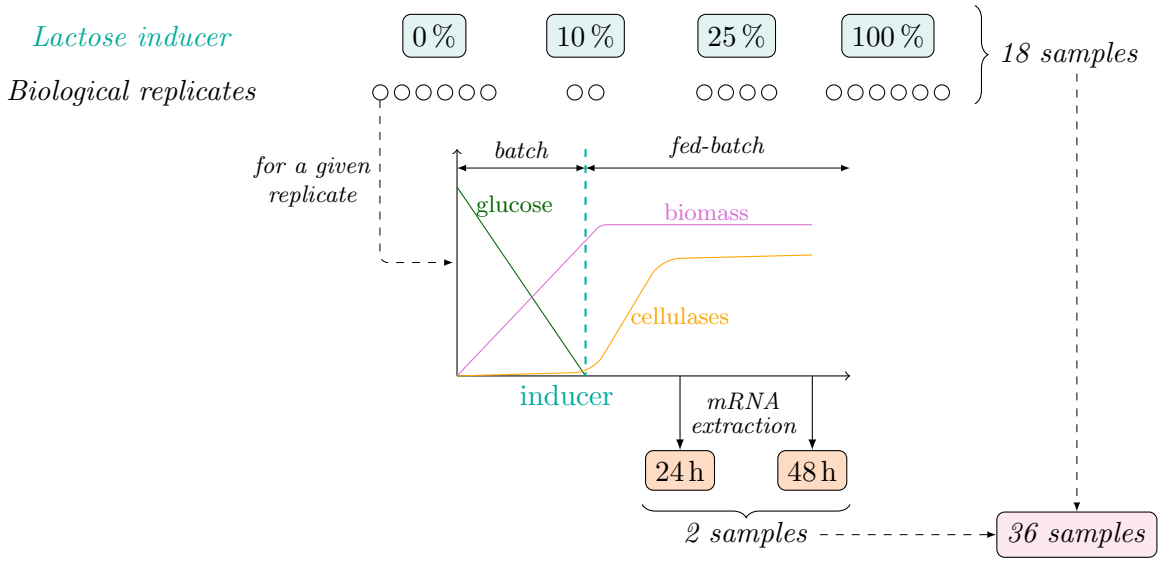


Figure 3.6 ~ EXPERIMENTAL DESIGN FOR RNA-SEQ DATA ~
*Experiments are designed for refining cellulase production knowledge.*Trichoderma reesei `Rut-C30` *firstly grown on glucose in a batch mode. Cellulase production induction is performed with various lactose concentrations in a fed-batch mode. RNA is extracted* 24 h *and* 48 h *after start induction and used for RNA-seq experiments.*

edge about cellulase production in the `Rut-C30` strain. For this purpose, we focus on both the repressor and the inducer effects. Indeed, glucose and lactose are respectively known to be a repressor and an inducer to cellulase production in *T. reesei*. We expect additional transcriptomic discoveries by varying the repressor/inducer concentrations. RNA-seq data are thus generated using transcriptomes of `Rut-C30` at 24 h and 48 h when it is cultivated on various culture media which contain different concentrations of a glucose/lactose mix. The chosen mixtures follow these glucose/lactose proportions: 100 %/0 %, 90 %/10 %, 75 %/25 %, and 0 %/100 %. Taking into account biological replicates, the 9129 genes of *T. reesei* are evaluated through 36 samples. The described experimental design is illustrated in Figure 3.6. From the generated RNA-seq data, pre-processing steps mentioned in Section 2.3 are performed and 650 genes including 21 TFs are selected. After an additional filtering, the final dataset used for network inference task is thus composed of 593 genes and 32 samples.

In order to evaluate the added value of the proposed network inference methods, comparisons to state-of-the-art methods have to be performed. In the following, we briefly describe the methods and the methodology used to carry out these comparisons.

~ State-of-the-art methods for comparative performance ~ This thesis aims at developing new approaches to infer reliable GRNs. In addition to assessing the behavior of the proposed methods themselves, comparisons to sate-of-the-art methods are also required. As mentioned in Section 2.4 and detailed in Chapters 4 to 6, *BRANE Cut*, *BRANE Relax* and *BRANE Clust* all

focus on the edge selection task performed on a fully-connected gene network weighted by gene-gene interaction scores and they yield sparse binary-valued networks. In this context, a pertinent evaluation consists, for a given complete weighted network, in comparing the classical edge selection task to the proposed ones which integrate biological and/or structural *a priori*. The chosen strategy is thus to firstly compute gene-gene interaction scores with state-of-the-art methods before to proceed to the edge selection task either by the classical thresholding or by our proposed approaches. The resulting binary-value networks are then compared using the methodology described in Section 3.2.2.

The choice of the method used to compute gene-gene interaction scores may appear insignificant, mistakenly. Indeed, as briefly mentioned in Section 2.4, both the classical and the proposed edge selection strategies have in common to favor strongly weighted edges. Unfortunately, in the context of Gaussian Graphical Models (GGM), edge weights are not absolute and their importance depends on the number of neighboring nodes. For a given edge $e_{i,j}$, a low value may be largely significant if corresponding nodes i and j take part in a highly connected group of nodes while the same low value can appear insignificant if the connected nodes are isolated. In such a case, as no restriction is made on the size of neighboring in both the classical and proposed edge selection strategies, using weights from GGM may thus arm the inference process. In addition, the general class of Bayesian models uses Bayesian criteria such as the Akaike Information Criterion (AIC) (Akaike, 1974) or the Bayesian Information Criterion (BIC) (Schwarz, 1978) for instance, to select the best parameters for their models, directly yielding a subset of selected edges and so the GRN. Therefore, we perform our comparison using weights from two top-performing methods: CLR (Faith *et al.*, 2007) and GENIE3 (Huynh-Thu *et al.*, 2010), computed from benchmark datasets previously presented. These two methods are frequently used as benchmarks (Meyer *et al.*, 2008; Zhang *et al.*, 2013; Roy *et al.*, 2013). Notably, GENIE3 was the best performer in the used DREAM4 and DREAM5 datasets. A post-processing step with Network Deconvolution (ND), developed by Feizi *et al.* (2013), was also used for a dual comparison. Firstly, ND is applied on CLR and GENIE3 weights leading to corrected ND-CLR and ND-GENIE3 weights. From these weights, either the classical thresholding and the proposed edge selection strategy are computed and compared. A supplemental comparison can be performed between the classical thresholding on corrected weights and the proposed method on the uncorrected weights.

As introduced in Section 3.1.3, the edge selection task can be improved by integrating a gene clustering information. The proposed method *BRANE Clust* specifically deals with this concept, as detailed in Chapter 6. In such a case, it can also be relevant to compare the clustering results in addition to the inference results, as gene clusters may be more informative — although in a different way — than the network itself. Clustering-based comparisons was thus carried out against the state-of-the-art method WGCNA (Langfelder and Horvath, 2008). We also chose to compare our clustering results with those obtain by X-means (Pelleg and Moore, 2000). The latter, an extension to K-means (Steinhaus, 1956; MacQueen, 1967) with an optimal number of classes, is not specific to biological applications, yet was used recently (Wang *et al.*, 2012a; Halleran *et al.*, 2015) in this context. The methodology used to compare clustering results is described in Section 3.2.3.

Once GRNs are inferred for a given method, their evaluations require numerical and biological validation. For this purpose, we present the methodology used to validate inference results.

3.2.2 Inference metrics and databases

A comprehensive evaluation of GRN inference methods requires at least two levels of validation: numerical and biological. A numerical validation aims at comparing inferred networks to the reference one (true network) using performance metrics. We first present the two main approaches allowing to compute such performance metrics. In addition to this objective validation, a more complex evaluation based on biological knowledge has to be performed. Hence, we also present how to biologically validate inferred networks.

~ *Numerical validation* ~ In order to objectively compare performances of GRN inference methods, a numerical validation is used. It consists in comparing the inferred network to a reference one using well-defined performance metrics. Before detailing performance metrics, it is necessary to introduce some basics for network comparison. GRN inference task can be seen as a binary classification problem as the inference determine whether an edge is present or not in the final graph. Hence, in the inferred network $\mathcal{G}^*$, each edge $e_{i,j}$ is labeled by 1 if it is present and 0 otherwise. A similar labeling can be computed for the reference network $\mathcal{G}_r$ and a 2×2 confusion matrix is used to performed edge-to-edge comparisons between the inferred and reference networks, see Table 3.4. This leads to the identification of True Positive (TP), True Negative (TN), False Positive (FP) — type-I error — and False Negative (FN) — type-II error — edges.

		Label of edge $e_{i,j}$ in $\mathcal{G}^*$	
		Absence (0)	Presence (1)
Label of edge $e_{i,j}$ in $\mathcal{G}_r$	Absence (0)	TN	FP
	Presence (1)	FN	TP

Table 3.4 **~** 2×2 CONFUSION MATRIX **~**
This table is used for edge-to-edge comparisons between an inferred network $\mathcal{G}^*$ and a reference network $\mathcal{G}_r$.

For a given inferred network $\mathcal{G}^*$ — and so a given parameter λ — let |TP|, |TN|, |FP| and |FN| be the number of TP, TN, FP, and FN edges, respectively. Standard statistical measures can thus be computed such as: Precision P (3.10), Recall R (3.11) (or sensitivity or True Positive Rate TPR, False Positive Rate FPR (3.12), Specificity (3.13), and Accuracy (3.14).

$$P = \text{Precision} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}|}, \tag{3.10}$$

$$R = TPR = \text{Recall} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}|}, \tag{3.11}$$

$$FPR = \frac{|\text{FP}|}{|\text{FP}| + |\text{TN}|}, \tag{3.12}$$

$$\text{Specificity} = \frac{|\text{TN}|}{|\text{TN}| + |\text{FP}|}, \tag{3.13}$$

$$\text{Accuracy} = \frac{|\text{TP}| + |\text{TN}|}{|\text{TP}| + |\text{TN}| + |\text{FP}| + |\text{FN}|}. \tag{3.14}$$

Although simple, these generic statistical measures are the mostly used to compare gene networks (Butte and Kohane, 2000; Meyer *et al.*, 2008; Margolin *et al.*, 2006; Faith *et al.*, 2007). Computing these measures for various λ values generates a vector of performance metrics. Nevertheless, it is more convenient for comparison purpose, to summarize previous performance metrics into a single scalar instead of comparing vectors. Area under Receiver Operator Characteristics (ROC) curve (AUC) is a common way to sum up performances. The underlying curve represents the Recall R as a function of the False Positive Rate (FPR). However, Davis and Goadrich (2006) recommend to use Precision-Recall curve instead of ROC curves when biases occur in class distribution, as it is the case for GRN inference results. Hence, the area under Precision-Recall curve (AUPR) is preferred to sum up and compare performances of GRN inference methods.

How to construct ROC or PR curves? From a global view point, each point in the ROC or PR space represents a performance measure for a specific classifier. In our context, a classifier can be assimilated to a given edge selection. In other words, from a complete weighted gene network, various threshold parameters λ responsible for edge selection are employed. Thus, at each generated network, a point in the ROC or PR space is computed. Then a linear interpolation is performed and area under curves are approximated using trapezoidal areas created between consecutive ROC or PR points. Authors in Davis and Goadrich (2006) proposed to correct PR points to a better interpolated value. However, this correction is rarely used in practice. It can thus be obvious that the choice of the λ range is essential to rigorously compare methods. Identical range and precision for λ values have to be used for all comparisons.

To overcome the dependence on the threshold parameter, the DREAM project proposes to define differently performance metrics P, R and FPR used to construct ROC and PR curves (Prill *et al.*, 2010). From a complete weighted network composed of $G(G-1)$ directed edges, a descending ranked order edge list is obtained such that the first edge in the list is maximally weighted and the last one is minimally weighted. Then, instead of evaluating performance metrics at a given threshold, they evaluate performance metrics as functions of the cutoff $k \in \{1, \ldots, E\}$ in the edge-list. Thus, a given point in the ROC or PR space is based on the new quantities Precision(k), Recall(k) and $FPR(k)$, respectively defined as follow:

$$\text{Precision}(k) = \frac{\text{TP}(k)}{k}, \tag{3.15}$$

$$\text{Recall}(k) = \frac{\text{TP}(k)}{p}, \tag{3.16}$$

$$FPR(k) = \frac{\text{FP}(k)}{n}, \tag{3.17}$$

where $\text{TP}(k)$ and $\text{FP}(k)$ denote the number of TP and FP in the top k predictions of the edge-list, respectively. Quantities p and n denote the number of positive and negative edges in the gold standard, respectively. In practice, not all edges are evaluated and missing edges are added in random order at the end of the list. From these performance metrics, ROC and PR curves can be drawn and their respective area AUC and AUPR can be computed. However, this approach is not adapted for binary-valued networks and cannot be applied for edge selection method comparison. As the main contribution of this thesis is the development of edge selection strategies, the firstly presented performance metrics (3.10) and (3.11) have been opted for and performances are numerically evaluated in terms of AUPR.

Supplemental measures exist to objectively compare gene networks. Emmert-Streib *et al.* (2012) expose ontology-based and network-based measures. Ontology-based measures allow to quantify the biological relevance of the inferred network by comparing groups of connected genes to known pathways identified in publicly available databases such as Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000) or Gene Ontology (GO) (Ashburner *et al.*, 2000). Ontology-based measures are rarely used in practice due to the usual lack of information regarding non-model organisms. Network-based measures explicitly consider network structure. For instance, Zhao *et al.* (2008) propose to combine the Dijkstra distance (Dijkstra, 1959) for type-I errors (false positive) while type-II errors (false negative) are weighted by one.

In addition to objective performance metrics, GRNs can also be evaluated in terms of biological relevance, especially for their predictive aspect. We now present some useful tools that can be used for evaluating biological relevance of a GRN inferred by a given method.

~ *Biological validation* ~ GRNs are built on the idea of extracting novel or finer information on gene regulation mechanisms. Thus, an expert validation and analysis of the GRN are preferably required to assess the ability of a GRN inference method to provide useful biological insights. Expert analysis brings into play several tools and databases acting at various scales. Biological validation is preferentially performed on network inferred from real benchmark datasets of model organisms such as *Escherichia coli* or *Saccharomyces cerevisae*. For these two micro-organisms, we have at your disposal well-filled databases such as, for instance, RegulonDB (Gama-Castro *et al.*, 2016) or EcoCyc (Keseler *et al.*, 2013) for *E. coli* and SGD (Saccharomyces Genome Database) (Cherry *et al.*, 2012) for *S. cerevisae*. These databases contain validated and predicted gene information that can be used to help us in assessing the coherence of the inferred networks. For less known species, an expert, rigorous and intensive bibliographic study has to be carried out through the literature, to identify recently unveiled interactions, possibly on related

species.

Networks can also be evaluated more locally. For this purpose, we study the biological relevance of modules, defining here by links between a given TF and their predicted targets. Modules can be evaluated through gene annotations, often coming from Gene Ontology (GO) database (Ashburner *et al.*, 2000). For a given gene, a Gene Ontology attribute categorizes for molecular function, cellular component and biological process. They can thus be employed to discover significant functional enrichment in modules or regulatory programs. Indeed, in a given module, statistical tests are used to analyze whether a function is predominant in the module when compared to the genome scale. For instance, if 10 genes are assigned to a given functional category at the genome scale, and 8 of these genes are present in the module, we may conclude to a significant enrichment of the module for the specific functional category. In such a case, a high confidence is attributed to the module.

As mentioned in Section 2.1, TFs contain a relatively well conserved DNA-binding site sequences allowing them to bind regulatory sequences of the gene to be regulated. The JASPAR database (Sandelin *et al.*, 2004; Mathelier *et al.*, 2013) gives a list of already identified DNA-binding sites for various organisms. If the TF binding site is known, the discovery of this pattern in the promoters of its predicted targets, given by the inferred network, may validate them. On the contrary, if the DNA-binding site is unknown, we may search for a consensus pattern into the promoters of its predicted targets. The discovery of such a consensus pattern in the $\overline{\text{TF}}$s promoters allows us to validate the fact that predicted targets are effectively regulated by the same TF. Hence, the identification of an enriched DNA-binding site pattern in a module can be in favor of its validation. Nevertheless, without additional experiments, the link between these predicted targets and the proposed TF cannot be — sensu stricto — validated. Indeed, to ensure that the discovery consensus pattern is related to the proposed TF, physical links between the TF and its predicted targets have to be highlighted. These physical interactions can be evaluated *via* ChipSeq experiments, for instance. Note that looking for an identified or a consensus pattern on the $\overline{\text{TF}}$s promoters requires a global or random search in order to evaluate the significance of the pattern discovery. For this purpose, the frequency of the pattern at the genome scale — on all the gene promoters — is statistically compared to those obtained at the module scale. Another strategy consists in using a subset of gene promoters, randomly chosen among all the gene promoters, instead of using all promoters. Several tools have been developed toward this kind of promoter analysis such as Regulatory Sequence Analysis Tools (RSAT) (Thomas-Cholier *et al.*, 2008), Multiple EM for Motif Elicitation (MEME) (Bailey *et al.*, 2006) or TOUCAN (Aerts *et al.*, 2003, 2005), for instance. Note that this validation approach can be limited as knowledge about binding site is too often poor for non-model micro-organisms, as it is the case for our fungus *Trichoderma reesei*.

Predictions between TFs and $\overline{\text{TF}}$s may also be evaluated thanks to the STRING database (Franceschini *et al.*, 2013). It references both known and predicted protein-protein interactions (direct or indirect) from 2031 organisms. Interactions are derived from five main sources: genomic context predictions, high-throughput experiments, (conserved) co-expression, automated text-mining and previous knowledge in databases. For some species, additional experiments,

such as double-hybrid or gene deletion/over-expression are also take into account to reference physical and genetic interactions, respectively. Several criteria address an evidence score suggesting a functional link: co-occurrence across genomes (Co-O), co-expression (Co-E), co-mentioned in PubMed abstracts (Co-M), neighborhood in the genome (N), gene fusion (F), experimental and biochemical data (E) and association in curated databases (Db). A combination of their respective probabilities, corrected from the chance of randomly observing an interaction, leads to a combined score (CS) per link (von Mering *et al.*, 2005). The above notions and abbreviations will be reused in Sections 4.1.3 and 6.3.3 dedicated to the validation of the *BRANE Cut* and *BRANE Clust* approaches, respectively. In a GRN validation context, a significant combined score between a TF and its target may be used to ascertain whether the predicted link has some biological relevance. In addition, even if links between $\overline{\text{TF}}$s do not exists in a GRN, it can be interested to consider CSs for couples of $\overline{\text{TF}}$s. Indeed, if high CSs are observed between $\overline{\text{TF}}$s of a given module, this suggests a co-expression of these $\overline{\text{TF}}$s leading to a higher confidence for the inferred module. Others databases gathering protein-protein interactions, pathway interaction, etc. can also be used for a complemental validation. Notably, we can cite KEGG (Kanehisa and Goto, 2000), DIP (Database of Interacting Proteins) (Salwinski *et al.*, 2004), PINA (Protein Interaction Network Analysis) (Cowley *et al.*, 2011) or IntAct (Orchard *et al.*, 2013).

Additional validation may be performed by analyzing regulatory pathways from other strains or phylogenetically close species. Indeed, these species often share genes having the same function and implied in the similar pathway. Hence, if a predicted link given by the GRN is unknown for the organism of interest, its presence in phylogenetically close species may be favorable to its validation. For instance, the tool FungiPath (Grossetête *et al.*, 2010) provides a large orthology database for various fungus species.

All of the above validations, sometimes fastidious, mostly provide hints and suggestions of plausible findings. They are somehow fragile. So finally, the last but not the less important validation, is to perform biological experiments. When the aforementioned analyses seem to give interesting new biological insights, the best way to validate them is to proceed to appropriate genetic engineering on studied cells. What we want to validate, in such a case, is the a posteriori quality of the a priori prediction regarding the implication of some TFs in a given phenotype. For this purpose, two kinds of experiments can be performed: knock-out — the deletion of the gene coding for the TF — and/or over-expression. The deletion of the TF allows us to obtain the direct effect of the TF on the phenotype. Indeed, if the TF is an activator, its deletion yield a lost or lessened phenotype. On the contrary, if the TF is a repressor, we expect an improved phenotype. Note that if the phenotype is similarly conserved, the deletion of the TF seems to have no effect, suggesting a bad prediction regarding the involvement of the TF in the phenotype. Gene over-expression experiments theoretically yield inverse conclusions. Nevertheless, as gene over-expressions essentially perturb the mechanism, conclusions regarding the resulting phenotype are less direct than with knock-out experiments.

Some of GRN inference methods can return gene clustering information for which specific metrics and databases are needed to evaluate the pertinence of such output. We thus now detail the methodology used to validate clustering results when available.

3.2.3 Clustering metrics and databases

Before to introduce the methodology for gene clustering evaluation, we firstly discuss their use on gene expression data and their incorporation in the context of GRN inference.

Clustering aims at partitioning a set of data into smaller subsets, called clusters, such that: *i*) within a given cluster, data are as similar as possible, *ii*) between clusters, data are as dissimilar as possible. Clustering techniques are usually performed on gene expression data to obtain group of genes having similar behavior across experimental conditions. For this purpose, K-means (Steinhaus, 1956; MacQueen, 1967), hierarchical clustering (Jain and Dubes, 1988; Kaufman and Rousseeuw, 2005), spectral clustering (Ng *et al.*, 2001), SOM (Kohonen, 2000) and Cluster Affinity Search Technique (CAST) (Ben-Dor *et al.*, 1999) are favorite approaches in biology. We refer to Zhang *et al.* (2004); de Souto *et al.* (2008) and Pirim *et al.* (2012) for a larger review and a comparison of clustering techniques used in a genomic context. In each generated cluster, genes are expected to share similar gene expression profiles. When it is the case, we can suppose that these genes can also share similar functionality or belong to the same pathway. Relevance of gene clusters is thus biologically evaluated through functional enrichment analysis, as mentioned in Section 3.2.2.

These clustering analysis are commonly performed in a separate manner from the GRN inference. However, the latter can benefit from gene clustering information as shown in some GRN methods combining clustering and inference. For this purpose, two main philosophies are employed. On the one hand, gene clusters are pre-computed and then used to drive the inference. On the other hand, inference and clustering are jointly constructed. Section 6.1 is dedicated to a deeper description of these related methods. However, we can discuss here the benefit of embedding gene grouping information into the inference. Assuming that the most reliable groups of genes reflect co-expressed genes, the reward reaped from considering such groupings may reveal threefold. First of all, this *a priori* could improve the detection of true interactions between a TF and its targets by enforcing co-expressed genes to be linked to their most probable TF. Then, clustering could help the identification of the underlying network structure and thus promote the modularity expected in GRNs. Finally, combinatorial regulation can be detected more easily as TFs acting together are expected to belong to the same cluster. From another viewpoint, clustering allows us to group similar genes, hence supposing the use of an appropriate similarity measure. This similarity measure can help us to refine or complement usual gene-gene interaction scores.

Dealing with clustering-based GRN inference approaches entails an evaluation of both the inferred network and the generated gene clusters. Since network evaluation was introduced in Section 3.2.2, we forthwith detail two approaches developed to compare two clustering results. Let $X = \{X_1, \ldots, X_M\}$ and $Y = \{Y_1, \ldots, Y_N\}$ be two partitions of G points. The Rand Index (RI) measure (Rand, 1971) denoted by I_r evaluates how two clustering results match by (3.18):

$$I_r = \frac{a+b}{a+b+c+d}, \tag{3.18}$$

where a denotes the number of pairs which are assigned to the same cluster in both X and Y, b

counts the number of pairs which are not assigned in the same cluster in X and in Y. Quantity c (resp. d) denotes the number of pairs which are in the same cluster in X but in different clusters in Y (resp. in different clusters in X but in the same cluster in Y). This measure has the main drawback to be non zero if two random partitions are evaluated. To overcome this problem, Hubert and Arabie (1985) propose an adjustment by rescaling the raw Rand Index according to the one computed from two random partitions. Meilă (2003) introduces the Variation of Information (VI), denoted by I_v, a measure related to the mutual information which evaluates the distance between two partitions. It is defined by:

$$I_v = -\sum_{i,j} r_{i,j} \left(\log\left(\frac{r_{i,j}}{p_i}\right) + \log\left(\frac{r_{i,j}}{q_j}\right) \right), \tag{3.19}$$

where $p_i = |X_i|/G$ and $q_j = |Y_j|/G$. These two quantities correspond to the proportion of elements in the cluster i of the partition X and the proportion of the elements in the cluster j of the partition Y, respectively. The term $r_{i,j} = |X_i \cap Y_j|/G$ is the proportion of elements assigned to the cluster i of the partition X and assigned to the cluster j of the partition Y.

While VI and RI are suitable measures for comparing two clustering, it can also be useful to assess the pertinence of single clustering, when no reference is available, for instance. In such a case, a single clustering can be intrinsically evaluated — to evaluate whether a better clustering could be obtained. For this purpose, the silhouette measure has been developed by Rousseeuw (1987). It tends to attribute a score $s_i \in [-1, 1]$ at each element i to be classified:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \tag{3.20}$$

where a_i is the average dissimilarity of i with other data in the same cluster and b_i is the lowest average dissimilarity of i to clusters different from the cluster of i. The cluster giving the lowest average dissimilarity is thus the nearest cluster for i, and it is called its neighborhood. A satisfactory assignment will be recorded if the score s_i is close to 1. An s_i value close to -1 indicates that the element i will be better classified in its neighborhood cluster. Finally, an intermediate score of 0 indicates that the element i is on the border between its own cluster and its neighborhood cluster. Averaging all scores s_i gives us an estimate of the global clustering correctness. The presented metrics are not restricted to gene clustering evaluation. Indeed, they have been conceived for generic clustering comparisons and can be applied in a large number of fields, such as image segmentation, see for instance Chapter 7.

Unlike for GRN evaluation, where a reference network is available, reference gene clustering is rarely available. Nevertheless, for bacteria species, we used the fact that they contain operons in their genome to overcome the lack of reference clustering. Operons are defined as a group of genes, adjacently located after a unique promoter, and thus subject to the same regulation. For bacteria species, we can thus compare clustering results with a reference constructed through the knowledge about operons. For the *Escherichia coli* dataset from DREAM5 (Section 3.2.1), cluster comparison is performed on two levels with VI: first, between different methods including WGCNA (Langfelder and Horvath, 2008) and X-means (Pelleg and Moore, 2000), and second

with the operon-based reference obtained from RegulonDB v.9.0 (Gama-Castro *et al.*, 2016).

After presenting state-of-the-art methods for GRNs inference and the methodology to evaluate them, we now introduce the main mathematical basics and optimization algorithms used to develop the method proposed in this thesis. Note that, in contrast to the previous sections — voluntary wordy — the following one may potentially appear heterogeneous due to the introduction of mathematical aspects. Nevertheless, this choice was opted for grouping all mathematical tools used in order to provide to the reader all the requirements needed for a better understanding of the proposed *BRANE* approaches.

3.3 Graph optimization and algorithmic frameworks

We thus now focus on the GRNs inference problem itself, and more precisely on the edge selection task. This thesis provides new methods to improve this selection. The proposed methods are developed in Chapter 4 (*BRANE Cut*), Chapter 5 (*BRANE Relax*) and Chapter 6 (*BRANE Clust*). They are based on an energy function, to be optimized, derived from the classical thresholding and require appropriate optimization algorithms for which theoretical aspects are provided in this section.

3.3.1 Optimization view point for edge selection

As mentioned in Section 2.4, an inference problem can be directly solved from a node-valued graph, where the node i is multi-valued by the gene expression profile $\mathbf{m}_i$. However, it can be more convenient to deal with edge-valued graph where edge weights correspond to gene-gene interaction scores, as for the majority of introduced methods in Section 3.1. We thus now focus on these edge-valued graphs for which an edge selection is needed to select plausible regulatory links only. Figure 2.11 - p. 34 recalls this process. Although gene-gene interaction scores play a central role in obtaining reliable GRNs, edge selection constitutes a crucial step. The selection of relevant edges is classically done by simply removing all edges whose (absolute) weights are lower than a threshold λ. Edge selection in classical thresholding can be viewed as a binary edge classification. It can be formulated as a trivial optimization problem. It allows the integration of biological and structural *a priori* towards GRN result refinements introduced in the next chapters of the present thesis.

We recall that, from a complete undirected weighted graph $\mathcal{G}\ (\mathcal{V},\ \mathcal{E};\ \omega)$, the inferred GRN is denoted by $\mathcal{G}^*$ and the corresponding set of present edges $\mathcal{E}^*$. Let us define, for each edge $e_{i,j} \in \mathcal{E}$ with weight $\omega_{i,j}$, a binary label $x_{i,j}$ of edge presence such that:

$$\forall (i,j) \in \mathbb{V}^2 \quad \text{and} \quad j > i, \qquad x_{i,j} = \begin{cases} 1 & \text{if } e_{i,j} \in \mathcal{E}^*, \\ 0 & \text{otherwise.} \end{cases} \tag{3.21}$$

Each label $x_{i,j}$ indicates the presence or the absence of the edge $e_{i,j}$ in the final graph $\mathcal{G}^*$. Performing a classical thresholding to select relevant edges is equivalent to defining an optimal

edge labeling $x^*_{i,j}$ such that:

$$\forall (i,j) \in \mathbb{V}^2 \quad \text{and} \quad j > i, \qquad x^*_{i,j} = \begin{cases} 1 & \text{if } \omega_{i,j} > \lambda, \\ 0 & \text{otherwise.} \end{cases} \tag{3.22}$$

This optimal edge labeling $\boldsymbol{x}^*$ can be obtained by solving a simple regularized optimization problem:

$$\underset{\boldsymbol{x} \in \{0,1\}^E}{\text{maximize}} \quad \sum_{\substack{(i,j) \in \mathbb{V}^2 \\ j>i}} \omega_{i,j}\, x_{i,j} \quad + \quad \lambda\,(1 - x_{i,j}), \tag{3.23}$$

where E is the number of edges and equals $G(G-1)/2$ as the graph $\mathcal{G}$ is supposed undirected and λ the regularization parameter. The first term alone would select all edges. The second term restricts this selection to those with weights larger than λ. Hence, the threshold parameter λ in classical thresholding becomes a regularization parameter.

Why does (3.22) solve Problem (3.23)? For a given edge $e_{i,j}$, the function to be maximized is $f(x_{i,j}) = \omega_{i,j}x_{i,j} + \lambda(1 - x_{i,j})$. As $x_{i,j}$ is a binary label, the function f takes two values only: $f(0) = \lambda$ and $f(1) = \omega_{i,j}$. If $\omega_{i,j} > \lambda$, the label $x_{i,j}$ maximizing f should be equal to 1 and conversely, if $\omega_{i,j} \leq \lambda$, the label $x_{i,j}$ which maximizes f has to be 0. This is exactly what we obtain when we apply a classical thresholding for edge selection.

In (3.23), the two terms depending on $x_{i,j}$ are complementary. Inverting this complementarity allows us to re-express Problem (3.23) into the following minimization problem:

$$\underset{\boldsymbol{x} \in \{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j) \in \mathbb{V}^2 \\ j>i}} \omega_{i,j}\,(1 - x_{i,j}) \quad + \quad \lambda\, x_{i,j}, \tag{3.24}$$

for which the explicit form in (3.22) is recovered. Indeed, by analogy with the previous explanation, for a given label $x_{i,j}$, the f function takes two values only: $f(0) = \omega_{i,j}$ and $f(1) = \lambda$. Thus, to minimize f, the label $x_{i,j}$ has to be equal to 1 if $\omega_{i,j} > \lambda$ and 0 otherwise.

As for a large number of regularized problems, finding the regularization parameter is not trivial. In our cases, lowering λ increases the potential of recovering known gene interactions. However, unassisted threshold selection may unveil an excessive number of false positives in the GRN. To limit the selection of false positive edges, additional regularization encoding biological and/or structural *a priori* can be integrated to (3.23), or equivalently to (3.24). This strategy was employed in the developed *BRANE* approaches detailed in Chapters 4 to 6. Each formalized problem, integrating particular *a priori*, is solved with the appropriate algorithm, as it belongs to a particular class of optimization problem:

- ⋆ *BRANE Cut*: discrete optimization on binary edge labels solved using a *maximal flow* algorithm (Chapter 4),
- ⋆ *BRANE Relax*: continuous optimization on edge variables solved using *proximal* methods and acceleration tricks relying on *majorize-minimize* principle and block coordinate strategy (Chapter 5),

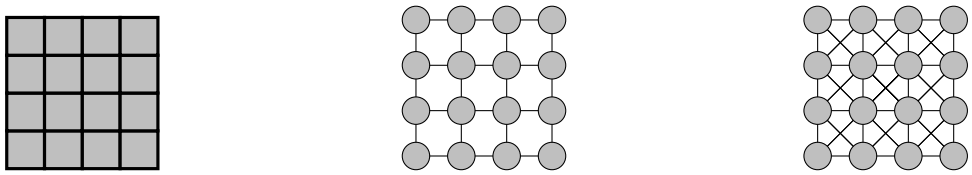
- ⋆ *BRANE Clust*: discrete and continuous optimization for edge and node variables, respectively, solved using an alternating optimization involving the *random walker* algorithm (Chapter 6).

The chosen algorithms — maximal flow (MF), proximal method (PM) and random walker (RW) — are popular in image processing and computer vision communities. Adaptation of these tools to unstructured networks, such as GRNs, are provided in each respective chapter. In the following, we give a theoretical framework about MF, PM and RW, in an image processing context, where images can be viewed as structured and regular graphs. In addition, a brief introduction to the Majorize-Minimize (MM) principle is also given.

3.3.2 Maximal flow for discrete optimization

A classical problem encountered in computer vision is image segmentation, which aims at partitioning an image (set of pixels) into multiple objects (subsets of pixels) sharing the same characteristics. In other words, image segmentation aims at assigning a label to each pixel in an image. Pixels sharing certain characteristics (intensity, color, texture, etc.) are assigned to the same label. A large number of approaches exists under various frameworks: thresholding and clustering, variational methods, graph partitioning methods, to name a few. Graph partitioning methods encompass several approaches such as: normalized cuts (Shi and Malik, 2000), random walker (Grady, 2006), minimum spanning tree (Meyer, 1994) or minimum cut (Ford and Fulkerson, 1956), for instance. We now focus on minimum cut models and algorithms proposed to solve them.

An image can be seen as a structured graph, where pixels of the image are associated to nodes. A node is classically linked by edges to its four nearest neighbors, corresponding to its four "nearest" pixels in the cardinal directions. A variety of more complex graphs exists and allows connections with more "nearest" neighbors such as an eight nearest neighbors structure, for instance. Figure 3.7 illustrates possible graph constructions from an image.



(a) Image with 4×4 pixels. *(b) 4-connected pixel graph.* *(c) 8-connected pixel graph.*

Figure 3.7 ~ Graph representations of a 4×4 image ~

Weights $\omega_{i,j}$ can be defined for each edge $e_{i,j}$, and are commonly related to pixel intensities I_i and I_j (Unger *et al.*, 2008):

$$\omega_{i,j} = \exp(-\beta(I_i - I_j)^2). \tag{3.25}$$

From the graph represented in Figure 3.7(b), two special nodes are added: the source s — node without entering edges — and the sink t — node without leaving edges. The new generated

graph is called a flow network $\mathcal{G}_f$ (or transportation network), and edge weights are called capacities. In a flow network $\mathcal{G}_f$, a cut is defined as a node partition into two disjoint subsets O and B such that $s \in O$ and $t \in B$ (subsets names borrow from image processing, denoting object and background). The capacity of a cut is obtained by summing the capacities of edges crossing the cut. As Figure 3.8 illustrates with a toy example, the minimum cut problem is thus to find an $s-t$ cut in $\mathcal{G}_f$ that minimizes the cut capacity.

(a) Cut capacity = 21. *(b) Cut capacity = 15.*

Figure 3.8 ~ CUTS IN A TRANSPORTATION NETWORK ***~***
(a) Arbitrary cut and (b) minimum cut in a flow network $\mathcal{G}_f$. The cut leads to a node partitioning such that the source s belongs to a subset O and the sink t to a subset B, for instance. The cut capacity is obtained by summing the weights of edges crossing the cut. Finding the minimal cut capacity solved the minimum cut problem.

From a mathematical viewpoint, the minimum cut problem can be viewed as a discrete optimization problem. Indeed, this problem aims at finding a label variable x_i for each node v_i, where the label reflects the class the node belongs to. As the basic problem implies two classes only, x_i is a binary variable taking 1 or 0 values. Two nodes are linked by an edge with a capacity $\omega_{i,j} \geq 0$. These capacities reflects pixel similarity in terms of intensity, color or texture, for instance, and drive the partitioning. For seeded image segmentation (Boykov and Jolly, 2000), a constraint is added on specific nodes s and t such that s belongs to one class and t to the other class. This constraint is equivalent to fixing x_s, the label of s, to 1 and to fixing x_t, the label of t, to 0. Hence, the minimum cut problem is thus simply formulated as the minimization of a discrete energy function:

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}\,|x_i - x_j|, \\ &\text{subject to } x_s = 1 \text{ and } x_t = 0. \end{aligned} \tag{3.26}$$

It has been proved in Ford and Fulkerson (1956) that a dual problem exists and consists in maximizing a flow from s to t in $\mathcal{G}_f$. The duality minimum cut/maximal flow is exploited for image segmentation in an approach called "Graph Cuts" in the computer vision community.

In a transportation network $\mathcal{G}_f$, a flow is a function assigning to each edge a value under two conditions. The first one is a capacity constraint: $f(e_{i,j}) \leq \omega_{i,j}$: for a given edge $e_{i,j}$, the assigned value of the flow $f(e_{i,j})$ is lower or equals to the edge capacity $\omega_{i,j}$. When $f(e_{i,j}) = \omega_{i,j}$, the edge is said saturated. The second condition refers to a divergence-free constraint: $f_e(v_i) = f_l(v_i)$. The sum of the flow entering each node v_i, and denoted by $f_e(v_i)$, is equal to $f_l(v_i)$, the sum of the flow leaving the node v_i. Hence, the problem consists in finding the maximal flow that going from s to t under the two mentioned constraints (Ford and Fulkerson, 1956). The resulting maximum flow value is equal to the capacity of the minimum cut. A large number of maximal flow algorithms have been proposed to solve the minimum cut problem (Edmonds and Karp, 1972; Goldberg and Tarjan, 1986; Boykov and Kolmogorov, 2004).

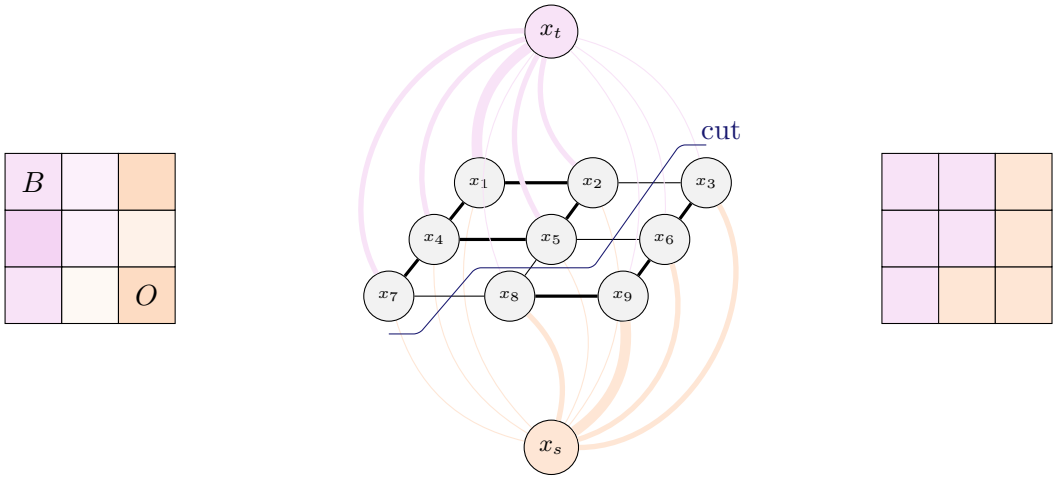
How Graph cuts can lead to a segmented image? Suppose that we aim at partitioning an image into two groups of pixels according to their intensities such that one group is related to the background B and the other group to an object O in the image. Thus, each pixel node v_i can be labeled by x_i and can either take 0 or 1 valuation. A node label of one corresponds to a pixel belonging to the object while a label of zero is for a pixel belonging to the background. Now, let the transportation network $\mathcal{G}_f$ be the one that links all pixel nodes to s and t with infinite weights. Capacities between pixel nodes are weights $\omega_{i,j}$ defined as in (3.25), for instance. The source s is labeled by 1 (the reference label for the object) and the sink t by 0 (and is the reference label for the background). Looking for a minimum cut in such a graph is the same as finding a maximal flow. The maximal flow computation leads to saturated edges. Nodes v_i reaching node s without encountering saturated edges will be labeled by 1 as it is the label of s. Similarly, nodes v_i reaching node t *via* non-saturated edges will be labeled by 0, the label value of t. Resultantly, a label is affected at each node and reflect the groups of pixels it belongs to: nodes labeled by 1 encode pixels belonging to the object while nodes labeled by 0 encode pixels belonging to the background. Figure 3.9 illustrates image segmentation with Graph Cuts.

The energy function to be minimized in (3.26) is one of the many possible energy function that can be solved using Graph Cuts. The generic formulation of the energy function $E(\boldsymbol{x})$ to be minimized *via* Graph Cuts for pixel-labeling problem takes the following form (Kolmogorov and Zabih, 2004):

$$E(\boldsymbol{x}) = \sum_{i \in \mathbb{V}} D_i(x_i) + \sum_{\substack{(i,j) \in \mathbb{V}^2 \\ j > i}} V_{i,j}(x_i, x_j), \tag{3.27}$$

where the first term is a data fidelity term derived from observations and reflects the cost to assign the label x_i to the node v_i (pixel p_i). The second term is a pairwise penalization term promoting spatial smoothness and encodes the cost to assign labels x_i and x_j to the nodes v_i and v_j (pixels p_i and p_j), respectively.

We shall see, in Chapter 4, how *BRANE Cut* use Graph Cuts framework to an edge selection problem integrating biological *a priori* for edge selection refinement in a GRN context. Note that, previous uses in different domains of bioinformatics can be found in Parikh *et al.* (2012); Azencott *et al.* (2013); Sugiyama *et al.* (2014). Nevertheless, using different biological *a priori*, the discrete problem could be relaxed into a continuous optimization problem and the use of a



(a) Initial image with seeds. *(b) Cut in the flow network $\mathcal{G}_f$.* *(c) Segmented image.*

Figure 3.9 ~ IMAGE SEGMENTATION WITH GRAPH CUTS ~
A flow network $\mathcal{G}_f$ is constructed from the graph of the initial image (a): all pixel nodes are linked to a source s and a sink t. Some pixel nodes, called seeds, are pre-labeled either by O or B such that these nodes are associated to the object or the background in the segmented image. After computing a maximum flow in $\mathcal{G}_f$ (b), a node is labeled by the label of x_s whether the node can be reached from the source s through non-saturated edges (thick edges) or by the label of x_t in the contrary case. The final labeling $\boldsymbol{x}^\star$ leads to the segmented image (c).

new range of tools and algorithms has to be considered. For this purpose, we introduce in the next Section 3.3.3, essentials regarding the random walker algorithm used for clustering.

3.3.3 Random walker for multi-class and relaxed optimization

As mentioned in Section 3.3.2, image segmentation is a frequently encountered problem in computer vision. While extensions of Graph Cuts for multi-label problems can be used, another algorithm called random walker provides an alternative. Random walker is a semi-supervised graph partitioning algorithm. Based on a network $\mathcal{G}$, valued on its edges by weights $\omega_{i,j}$, and composed of a set $\mathcal{V}$ of G nodes, let us define by $\mathcal{V}_M$ a subset of K pre-labeled (marked/seeded) nodes. We can thus define the complementary subset of unlabeled nodes $\mathcal{V}_U$. Knowing the label of the nodes in $\mathcal{V}_M$, the random walker algorithm assigns a label to the remaining nodes in $\mathcal{V}_U$.

In more details, let $K \in \mathbb{N}$ be the number of possible label values of nodes from $\mathcal{V}_M$. In addition, let $y_i \in \{1, \ldots, K\}$ be the label variable for node v_i and $\boldsymbol{y} \in \mathbb{N}^G$ be the vector gathering the label variables of the G nodes. Defining a cost function $E(\boldsymbol{y})$ as follow

$$E(\boldsymbol{y}) = \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}(y_i - y_j)^2, \tag{3.28}$$

the random walker algorithm solves the following constrained minimization problem:

$$\begin{aligned} &\underset{\boldsymbol{y}}{\text{minimize}} \quad E(\boldsymbol{y}), \\ &\text{subject to } y_i = k, \quad \forall v_i \in \mathcal{V}_M. \end{aligned} \tag{3.29}$$

In Grady (2006), the cost function $E(\boldsymbol{y})$ in (3.29) can be re-expressed as a combinatorial formulation of the Dirichlet integral:

$$E(\boldsymbol{y}) = \sum_{(i,j)\in\mathbb{V}^2} \omega_{i,j}(y_i - y_j)^2 = \boldsymbol{y}^\top L\, \boldsymbol{y},$$

where L is the combinatorial Laplacian matrix of the graph $\mathcal{G}$, defined as:

$$L_{i,j} = \begin{cases} d_i & \text{if } i = j, \\ -\omega_{i,j} & \text{if } v_i \text{ and } v_j \text{ are adjacent nodes,} \\ 0 & \text{otherwise,} \end{cases} \tag{3.30}$$

with d_i the degree of node v_i. Taking into account the constraint on pre-labeled nodes in (3.29), only the labels of unseeded nodes have to be determined, and the energy to be minimized in Problem (3.29) can be decomposed into:

$$E(\boldsymbol{y}_U) = \begin{bmatrix} \boldsymbol{y}_M^\top & \boldsymbol{y}_U^\top \end{bmatrix} \begin{bmatrix} L_M & B \\ B^\top & L_U \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_M \\ \boldsymbol{y}_U \end{bmatrix} = \boldsymbol{y}_M^\top L_M \boldsymbol{y}_M + 2\boldsymbol{y}_U^\top B^\top \boldsymbol{y}_M + \boldsymbol{y}_U^\top L_U \boldsymbol{y}_U, \tag{3.31}$$

where, in this context, $\boldsymbol{y}_M$ and $\boldsymbol{y}_U$ correspond to probability vectors of seeded and unseeded nodes, respectively. The unique critical point is obtained by differentiating the energy $E(\boldsymbol{y}_U)$ with respect to $\boldsymbol{y}_U$

$$\frac{\partial E(\boldsymbol{y}_U)}{\partial \boldsymbol{y}_U} = B^\top \boldsymbol{y}_M + L_U \boldsymbol{y}_U \tag{3.32}$$

thus yielding

$$L_U \boldsymbol{y}_U = -B^\top \boldsymbol{y}_M, \tag{3.33}$$

which is a system of linear equations with $|\mathcal{V}_U|$ unknowns. Note that if the graph is connected or if every connected component contains a seed, then (3.33) will be nonsingular and a unique solution will be found.

Let us define the set of labels for the seed nodes as a function $Q(v_i) = k$, for all $v_i \in \mathcal{V}_M$, where $k \in \{1, \ldots, K\}$. For each label k, a vector of markers $M^{(k)}$ of size $|\mathcal{V}_M|$ can thus be defined such that, for each node $v_i \in \mathcal{V}_M$

$$m_i^{(k)} = \begin{cases} 1 & \text{if } Q(v_i) = k, \\ 0 & \text{if } Q(v_i) \neq k. \end{cases} \tag{3.34}$$

The marker matrix $M = [M^{(1)}, \ldots, M^{(K)}]$ thus gathers all the vector of markers. By analogy, let us define the matrix $Y = [Y^{(1)}, \ldots, Y^{(K)}]$, where for all $k \in \{1, \ldots, K\}$, the vector $Y^{(k)}$ is of

size $|\mathcal{V}_U|$. For each node $v_i \in \mathcal{V}_U$, the component $y_i^{(k)}$ denotes the probability for the node v_i to be assigned to the label k. Probabilities in Y are unknown and have to be computed. Based on Equation (3.33), they can be computed by solving the following system of linear equations:

$$L_U Y = -B^\top M. \tag{3.35}$$

This strategy is equivalent to solving K binary-labeling sub-problems instead of solving a K-labeling problem. Nevertheless, dealing with probabilities enforces a sum-to-one constraint for each node $i \in \{1, \dots, G\}$ i.e.

$$\forall i \in \{1, \dots, G\}, \qquad \sum_{k=1}^{K} y_i^{(k)} = 1. \tag{3.36}$$

This implies that only $K-1$ systems of linear equations must be solved. Once probabilities at each node and for each label are computed, a final labeling has to be assigned. For this purpose, the label given by the maximal probability is assigned to each node:

$$\forall i \in \{1, \dots, G\}, \qquad y_i^* = \arg \max_{k \in \{1,\dots,K\}} y_i^{(k)}. \tag{3.37}$$

An algorithm name not so innocuous... Figure 3.10 illustrates how the random walker algorithm can segment an image in 3 classes. It can be interesting to view how the random walker algorithm assigns an unseeded pixel to a label. Indeed, an elegant analogy can be draw. Given a weighted graph, if a a random walker leaving the pixel is most likely to first reach a seed bearing label s, assign the pixel to label s.

BRANE Clust uses the random walker algorithm in its optimization strategy. We shall detail in Chapter 6 how such a clustering algorithm can be used to improve the inference of a GRN from gene expression data. While Graph Cuts and random walker were developed for some clustering tasks, other strategies exist to solve more generic problems involving larger classes of functions. The class of proximal methods is a powerful one and the following section is dedicated to its introduction.

3.3.4 Proximal methods for continuous optimization

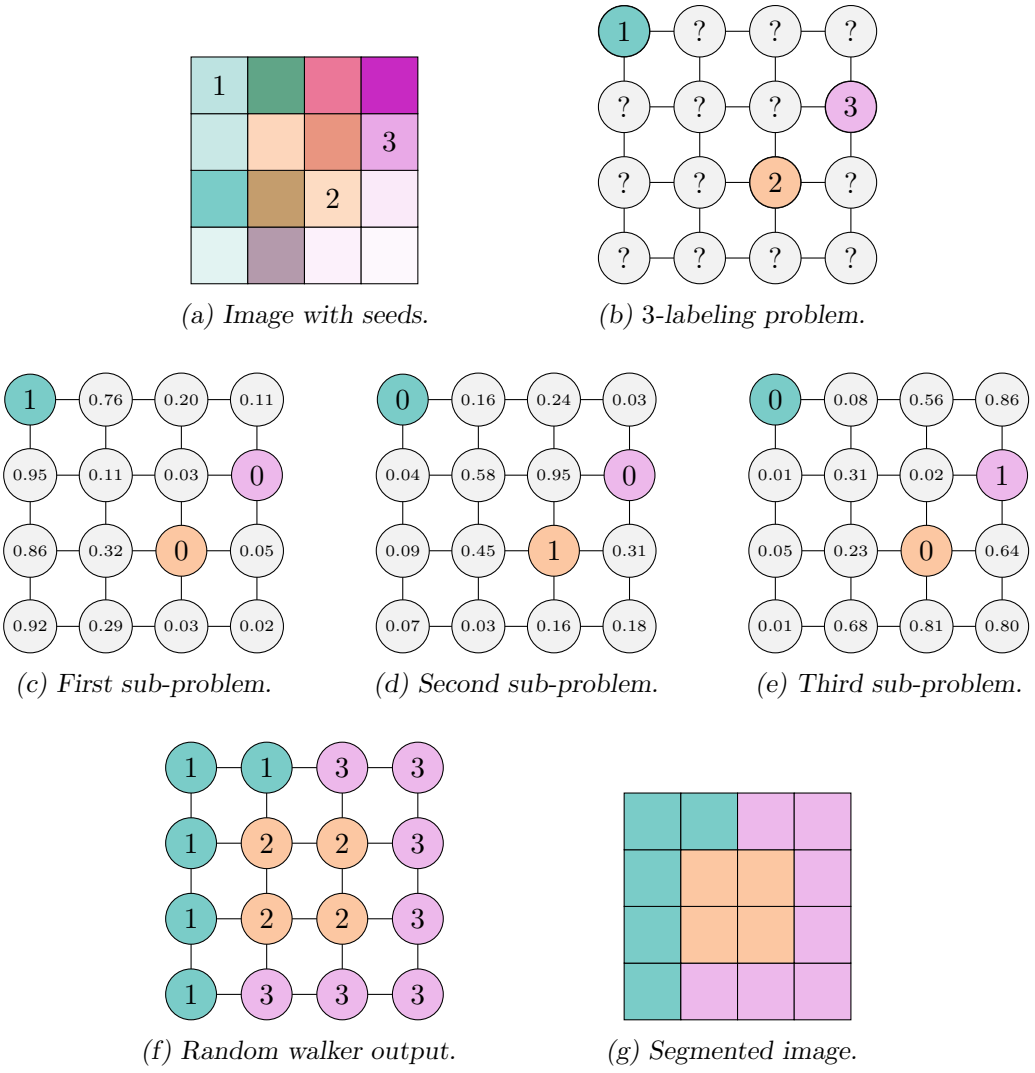
Proximal methods are used to solve continuous and convex optimization problems taking the following form:

$$\underset{x \in \mathbb{R}^N}{\text{minimize}} \quad f_1(x) + \cdots + f_n(x), \tag{3.38}$$

where functions $f_1, \dots, f_n$ are lower semi-continuous convex functions from $\mathbb{R}^N$ to $]-\infty, +\infty]$. Such kind of problems are encountered in constrained optimization, for instance, where one function usually encode data fidelity while other functions are added to encode some constraints.

What are convexity and its interest? We firstly introduce the basic for convex sets. Let x_1 and x_2 be two different points in $\mathbb{R}^N$. The line passing through x_1 and x_2 can be defined by points of the form (3.39):

$$y = \lambda x_1 + (1-\lambda) x_2, \tag{3.39}$$



(a) Image with seeds. *(b) 3-labeling problem.*

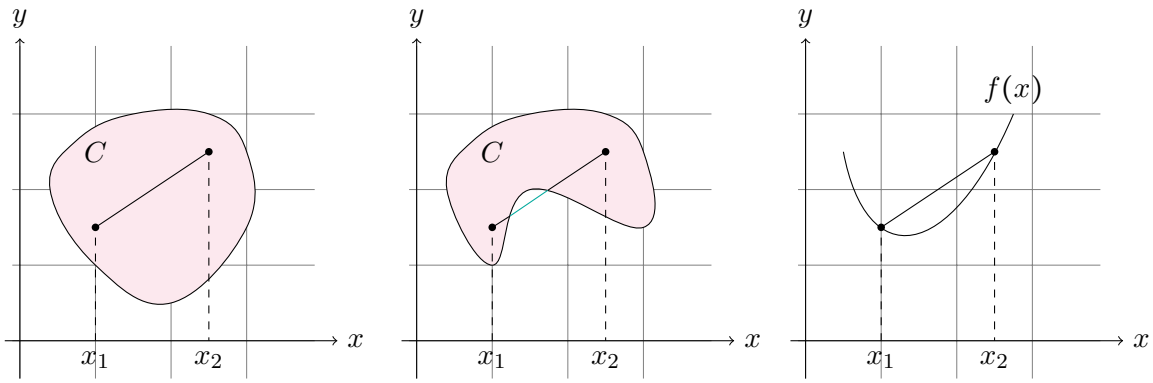
(c) First sub-problem. *(d) Second sub-problem.* *(e) Third sub-problem.*

(f) Random walker output. *(g) Segmented image.*

Figure 3.10 ~ IMAGE SEGMENTATION WITH RANDOM WALKER ~
On the initial image to be segmented (a), three labels valued by 1,2 and 3 are defined. The graph representation of the image is represented in (b), where edges are valued by weights from a function of the intensity gradient. The multi-labeling problem in (b) is decoupled into 3 sub-problems from (c) to (e), where for each of them, the label of the corresponding markers is set to 1 while keeping the others equal to 0. Assignment probabilities are then computed for the unlabeled nodes. They correspond to the probability that a random walker, starting at each node first reaches the pre-labeled node currently set to unity. The final graph partitioning in (f) is obtained by assigning to each node the label that corresponds to its greatest probability, yielding the segmented image (g).

where $\lambda \in \mathbb{R}$. Restricting $\lambda \in [0,1]$ reduces the passing line to the segment between x_1 and x_2. Now, for a given non-empty set C in $\mathbb{R}^N$, and two points $(x_1, x_2) \in C^2$, if the segment line defined

from (3.39) for $\lambda \in [0,1]$ is entirely contained is C, thus the set C is said convex. Otherwise, set C is termed non-convex. Figures 3.11(a) and 3.11(b) illustrate this property.



(a) A convex set C. *(b) A non-convex set C.* *(c) A convex function f.*

Figure 3.11 ~ CONVEX SET AND FUNCTION ~
Illustration of a convex set (a), a non-convex set (b) and a convex function (c).

Let C be a non-empty convex set on which the function f is defined. Let (x_1, x_1), two arbitrary points pertaining to the set C and $\lambda \in [0,1]$. The function f is convex if condition in (3.40) is verified:

$$\forall (x_1, x_2) \in C^2 \quad \text{and} \quad \forall \lambda \in [0,1], \quad f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2). \tag{3.40}$$

In other words, f is convex if the segment line between x_1 and x_2 — which corresponds to the chord from x_1 to x_2 — lies above the graph of f. Such a function is illustrated in Figure 3.11(c). If a strict inequality holds in (3.40), the function f is said strictly convex. An interesting property ensues from convex functions: local and global minima are confounded. Furthermore, a strictly convex function has at most one minimizer. In addition to the convexity, we introduce two other definitions pertaining to useful properties for the studied functions. Firstly, a function is said *proper* when the domain of the function is non-empty. Secondly, a function is characterized as *lower semi-continuous* if, for all sequence $(x_n)_{n \in \mathbb{N}}$ converging to a point x, $f(x) \leq \lim_{x_n \to x} \inf f(x_n)$. The notation $\Gamma_0(\mathbb{R}^N)$ refers to the set of proper, lower semi-continuous and convex functions.

In our framework, we reduce (3.38) to the particular case of a sum of only two functions:

$$\underset{x \in \mathbb{R}^N}{\text{minimize}} \quad f_1(x) + f_2(x), \tag{3.41}$$

where one of the two functions is necessarily differentiable. Let us assume that f_2, in addition to be convex, satisfies the differentiability assumption and has a β-Lipschitzian gradient, where $\beta \in]0, +\infty[$ is a Lipschitz constant. Note that a function is differentiable with a β-Lipschitz continuous gradient ∇f_2 if condition (3.42) is verified, for $\beta > 0$:

$$\forall (x_1, x_2) \in \mathbb{R}^2, \qquad \|\nabla f_2(x_1) - \nabla f_2(x_2)\| \leq \beta \|x_1 - x_2\|. \tag{3.42}$$

Function f_1 is only supposed to belong to $\Gamma_0(\mathbb{R}^N)$. Problem (3.41) can be solved by defining its solution as the limit of a sequence constructed in an iterative manner. Authors in Combettes and Wajs (2005) show that a solution to Problem (3.41) can be obtained through a proximal framework. The resulting algorithm, known as *forward-backward* algorithm, constructs the sequence $(x_k)_{k\in\mathbb{N}}$, such that, for all $k \in \mathbb{N}$, iterates are defined as:

$$x_{k+1} = x_k + \lambda_k \left(\text{prox}_{\gamma_k f_1}\left(x_k - \gamma_k \nabla f_2(x_k)\right) - x_k\right), \tag{3.43}$$

where $\gamma_k \in]0, 2/\beta[$ correspond to step-size parameters and $\lambda_k \in]0, 1]$ to regularization parameters.

What is the proximity operator, and its effect on the forward-backward algorithm? We refer to Combettes and Pesquet (2011) and Parikh and Boyd (2013) for a tutorial introduction to proximal optimization. The proximity operator of a function $f \in \Gamma_0(\mathbb{R}^N)$ at point $u \in \mathbb{R}^N$, denoted by $\text{prox}_f u$, was firstly introduced in Moreau (1965). It is defined as the unique minimizer of $f + \|\cdot - x\|^2$, or equivalently:

$$\text{prox}_f u = \arg\min_{x\in\mathbb{R}^N} \; f(x) \; + \; \frac{1}{2}\|u - x\|^2. \tag{3.44}$$

It generalizes the notion of projection onto a non-empty closed convex subset C of $\mathbb{R}^N$. Indeed, let us introduce the particular case where the function f is the indicator function ι_C of the non-empty closed convex set C of $\mathbb{R}^N$:

$$\forall x \in \mathbb{R}^N, \quad \iota_C(x) = \begin{cases} 0 & \text{if } x \in C \\ +\infty & \text{otherwise.} \end{cases} \tag{3.45}$$

The proximity operator simply becomes a projection operator onto the convex set C, denoted by P_C:

$$\begin{aligned} \text{prox}_{\iota_C} u &= \arg\min_{x\in\mathbb{R}^N} \; \frac{1}{2}\|u - x\|^2 + \iota_C \\ &= \arg\min_{x\in C} \; \frac{1}{2}\|u - x\|^2 = P_C(u). \end{aligned}$$

In such a context, taking f_1 as the indicator function of a non-empty closed convex subset C of $\mathbb{R}^N$ and f_2 as a convex and differentiable function with a β-Lipschitzian continuous gradient, problem (3.41) can be equivalently re-expressed as:

$$\underset{x\in\mathbb{R}^N}{\text{minimize}} \quad f_2(x) + \iota(x) \;=\; \underset{x\in C}{\text{minimize}} \quad f_2(x) \tag{3.46}$$

Iterations (3.43) in the forward-backward algorithm are thus reduced to

$$\forall k \in \mathbb{N}, \quad x_{k+1} = P_C\left(x_k - \gamma_k \nabla f_2(x_k)\right), \tag{3.47}$$

where $\gamma_k \in]0, 2/\beta[$. The resulting simplified algorithm is known as the *projected gradient algorithm* and it is useful in a large number of signal processing applications.

The latter algorithm — the projected gradient — is used in the developed method *BRAINE Relax*. Its direct application is detailed in Chapter 5. It is complemented to the Majorize-Minimize (MM) method, for which we introduce concepts in the following section.

3.3.5 Majorize-Minimize (MM) method

The Majorize-Minimize strategy was firstly introduced by Ortega and Rheinboldt (1970) to solve the minimization of a differentiable function f, in an iterative manner. Hunter and Lange (2004) provide a tutorial on MM principle and algorithms. At each iteration $k \in \mathbb{N}$, instead of minimizing the function f, the MM algorithm minimizes a *majorant* function $\mathcal{F}$ of f, leading to the following iterations, for all $k \in \mathbb{N}$:

$$x_{k+1} = \arg\min_{x \in \mathbb{R}^N} \mathcal{F}(x, x_k). \tag{3.48}$$

Figure 3.12 illustrates the MM principle. The majorant $\mathcal{F}$ of f has to be defined such that conditions in (3.49) are verified:

$$\begin{cases} \forall (x, x_k) \in \left(\mathbb{R}^N\right)^2, & \mathcal{F}(x, x_k) \geq f(x), \\ \forall x_k \in \mathbb{R}^N, & \mathcal{F}(x_k, x_k) = f(x_k). \end{cases} \tag{3.49}$$

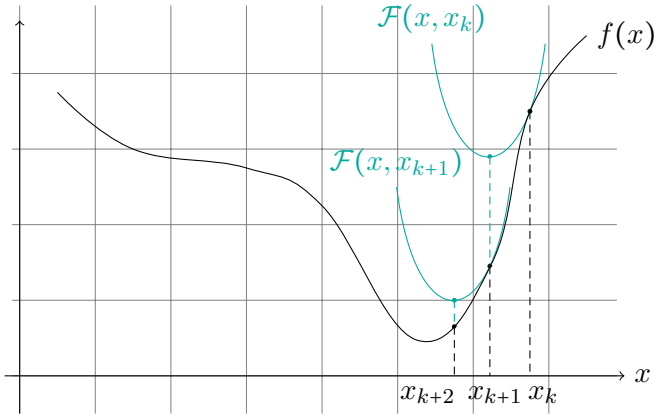


Figure 3.12 ~ MM PRINCIPLE ~ *At iteration k, a majorant $\mathcal{F}(x, x_k)$ is constructed such that $f(x_k) = \mathcal{F}(x_k, x_k)$. The argument x giving the minimum of the function $\mathcal{F}(x, x_k)$ is used to define the starting point at iteration $k+1$. Hence, the MM algorithm iteratively constructs a majorant, which is tangent to f at the current iteration point.*

In addition to the two required conditions on the majorant, such a majorant should be judiciously constructed in order to facilitate its minimization. Intuitively, a simple strategy relies on the following quadratic form of $\mathcal{F}$:

$$\mathcal{F}(x, x_k) = f(x_k) + (x - x_k)^\top \nabla f(x_k) + \frac{1}{2}\|x - x_k\|^2_{A_k}, \tag{3.50}$$

where, $\|\cdot\|_A$ is the weighted norm of $\mathbb{R}^N$ defined as:

$$\forall z \in \mathbb{R}^N, \quad \|z\|_A = \left(z^\top A z\right)^{\frac{1}{2}}, \tag{3.51}$$

with A a symmetric positive definite matrix. The minimum of such a majorant $\mathcal{F}$ defined in (3.50) can be explicitly obtained. Iterates in (3.48) can thus be re-expressed as:

$$\forall k \in \mathbb{N}, \quad x_{k+1} = \arg\min_{x \in \mathbb{R}^N} \mathcal{F}(x, x_k) = x_k - A_k^{-1} \nabla f(x_k) \tag{3.52}$$

Matrices $(A_k)_{k\in\mathbb{N}}$ are considered as preconditioning matrices. These choices drive the convergence speed of the MM algorithm. If the function f to be majorized is twice differentiable, the most efficient preconditioning matrix A_k, at each iteration $k \in \mathbb{N}$, appears to be $\nabla f^2(x_k)$, the Hessian of f at x_k. Nevertheless, as indicated in (3.52), MM iterations require the inversion of the matrix A_k. Choosing the preconditioning matrix as the Hessian can thus prejudice its inversion. Another potential problem is that choosing the inverse of the Hessian matrix as a preconditioner does not secure the convergence of the resulting Newton algorithm. In such a case, using an easily invertible approximation of $\nabla f^2(x_k)$ can turn out to be a judicious choice. This is especially the case when the approximation is a diagonal matrix (Chouzenoux *et al.*, 2014).

In Chapter 5, we shall detail how the *BRANE Relax* method we developed uses proximal methods presented in Section 3.3.4 and an MM strategy to refine a GRN.

This chapter was dedicated to the introduction of bases required to understand the methodology — summarized in Figure 3.13 — used to develop and evaluate the methods *BRANE Cut*, *BRANE Relax* and *BRANE Clust*.

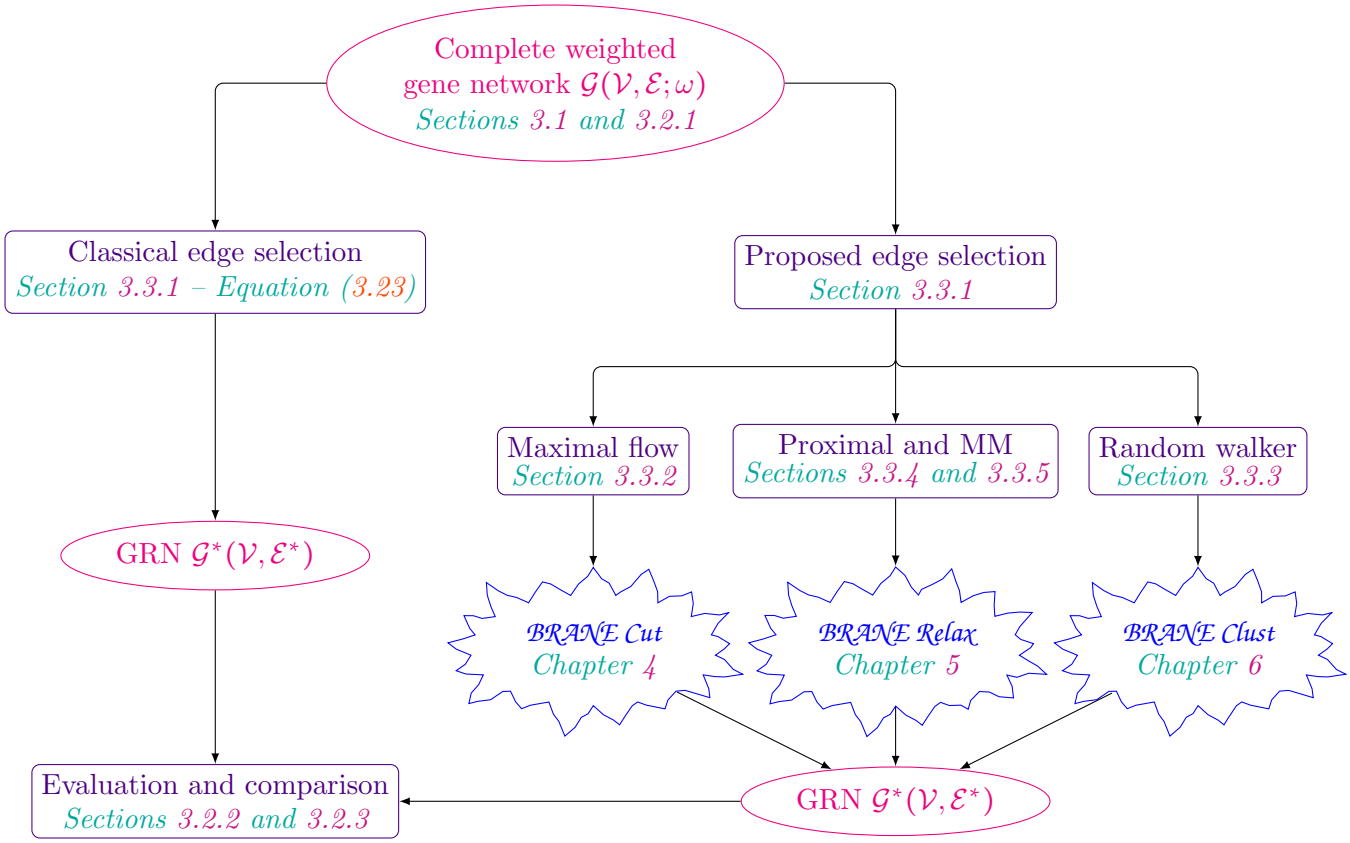


Figure 3.13 ~ Summing-up of notions introduced in Chapter 3 ~
Methodologies used to develop and evaluate GRN (Gene Regulatory Network) refinement methods BRANE Cut, BRANE Relax and BRANE Clust.

4

Edge selection refinement using gene co-regluation a priori (BRANE Cut)

"The world is continuous, but the mind is discrete."

David Mumford

This chapter is dedicated to the detailed presentation of BRANE Cut, a first contribution, published in Pirayre et al. (2015a). It is designed to be applied on complete graph for edge selection refinement in a GRN context. The proposed formulation integrates gene co-regulation as biological a priori *and takes the form of a minimum cut problem. The optimal solution is obtained by applying the maximal flow algorithm on the underlying transportation network. Promising results on benchmark datasets from DREAM4 and DREAM5 are presented as well as comparisons to state-of-the-art methods yielding maximal improvements reaching about* 14 %. *Finally, results on its application to gene prediction on a real dataset of* Trichoderma reesei *are also given (Pirayre et al., 2018b).*

Contents

As introduced in Section 3.3.1, our objective is to design a cost function depending on binary variables $x_{i,j}$ reflecting the presence/absence of the edge $e_{i,j}$ in the GRN. Such a cost function is based on the optimization formulation of the classical thresholding (CT) in (3.24), the expression of which is recalled:

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j)\in\mathbb{V}^2 \\ j>i}} \omega_{i,j}\,(1-x_{i,j}) \quad + \quad \lambda\, x_{i,j}, \tag{4.1}$$

In *BRANE Cut*, we take advantage of the availability of transcription factors (TFs) knowledge. A list of TFs often results from the combination of dedicated experiments to identify TFs and a knowledge of the literature. Moreover, some yet unvalidated TFs are also predicted as such thanks to the presence of specific DNA-binding motif in their sequence. Such a list is imperfect (oversight and wrong predictions) and its use as *a priori* may encourage us to carefully interpret the results. Up to a re-indexing of the list of genes, we suppose that the TFs are indexed first. Let $\mathcal{T} = \{v_1, \ldots, v_T\}$, be the set of nodes corresponding to TFs only, where T is the number of TFs and thus $\mathcal{T} \subset \mathcal{V}$. We introduce $\mathbb{T} = \{1, \ldots, T\}$ as the set of TF indices and by analogy $\mathbb{T} \subset \mathbb{V}$.

4.1 *BRANE Cut*: gene co-regulation *a priori*

4.1.1 Biological *a priori* and problem formulation

In addition to selecting strongly weighted edges as in the case of thresholding, *BRANE Cut* integrates two kinds of biological *a priori*. The first one is related to the differential connectivity of TFs, which can be observed in real GRN. The second one refers to an assumption made on a particular gene regulatory process. We shall detail the above *a priori* and their integrations in an energy functional to be optimized.

~ *TF-connectivity a priori* ~ Independently of the fact that TFs are less numerous than $\overline{\text{TF}}$s, regulatory relationships between couples of TFs are expected to be less frequent than between one TF and one $\overline{\text{TF}}$. This expectation may promote biological graphs with a modular structure (Chiquet *et al.*, 2009; Espinosa-Soto and Wagner, 2010) as illustrated in Figure 4.1.

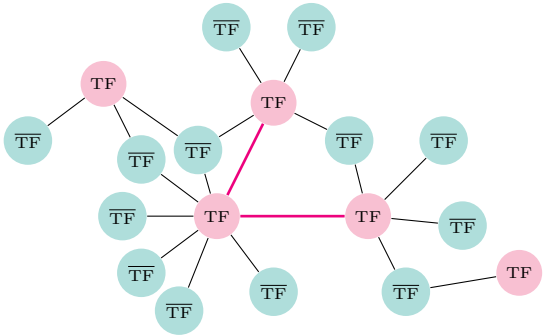


Figure 4.1 **~** TF-CONNECTIVITY *a priori* **~**
Pink edges link TFs *(pink nodes) and edges between one* TF *and one* $\overline{\text{TF}}$ *(green node) are drawn black. While* 27 % *of the genes are* TFs*, only* 10 % *of the interactions are between* TFs*.*

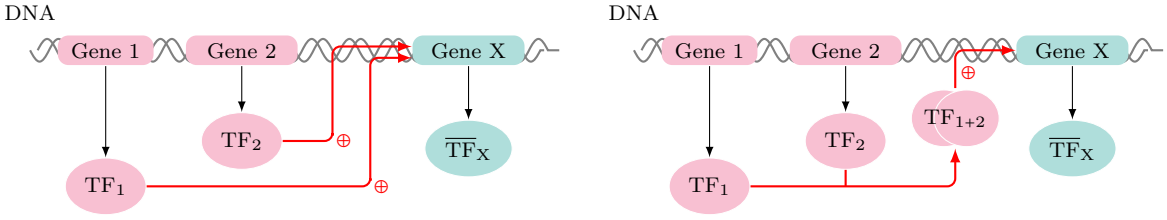
As we are looking for gene regulatory knowledge, we infer edges linked to at least one TF. In addition, based on our *a priori*, we recall that TF-$\overline{\text{TF}}$ edges could be preferentially preserve over TF-TF links. The proposed edge selection is driven by positive weights $\lambda_{i,j}$ which depend

on the three types of pairs of nodes i and j. We thus defined these case-dependent weights as follows:

$$\lambda_{i,j} = \begin{cases} 2\eta & \text{if } i \notin \mathbb{T} \text{ and } j \notin \mathbb{T}, \\ 2\lambda_{\mathrm{TF}} & \text{if } i \in \mathbb{T} \text{ and } j \in \mathbb{T}, \\ \lambda_{\mathrm{TF}} + \lambda_{\overline{\mathrm{TF}}} & \text{otherwise.} \end{cases} \tag{4.2}$$

Hence, $\overline{\mathrm{TF}}$-$\overline{\mathrm{TF}}$ edges have weights assigned to 2η, where η is a critical threshold. The parameter λ_{TF} acts in the neighborhood of a TF and is complemented by $\lambda_{\overline{\mathrm{TF}}}$ when the neighbor is a $\overline{\mathrm{TF}}$. They may be interpreted as two threshold parameters. This double threshold promotes grouping between strong and weaker edges among functionally-related genes. A similar approach is used in image segmentation (Canny, 1986) under the name of hysteresis thresholding to enhance edge connection and object detection with reduced sensitivity to irrelevant features (Ollion *et al.*, 2013). To promote TF-$\overline{\mathrm{TF}}$ interactions, the λ_{TF} parameter should be greater than $\lambda_{\overline{\mathrm{TF}}}$. To ensure that any TF involved interaction is selected first, we should verify that $\eta \geq \lambda_{\mathrm{TF}} \geq \lambda_{\overline{\mathrm{TF}}}$. Additionally, removing all $\overline{\mathrm{TF}}$-$\overline{\mathrm{TF}}$ edges amounts to setting their corresponding $x_{i,j}$ to zero. Consequently, η should exceed the maximum value of the weights ω. Since we address different data types and input weight distributions, we can easily renormalize them all to $\omega_{i,j} \in [0, 1]$, and choose 2η as the maximum value of weights i.e. $2\eta = 1$. When $\lambda_{\mathrm{TF}} = \lambda_{\overline{\mathrm{TF}}}$, no distinction is made between edge types. This is equivalent to using a unique threshold value, as in classical gene network thresholding. This can be interpreted as if, without further *a priori*, all genes were indistinguishable from putative TFs. However, different λ_{TF} and $\lambda_{\overline{\mathrm{TF}}}$ may be beneficial. For any fixed value of λ_{TF}, smaller values for $\lambda_{\overline{\mathrm{TF}}}$ improve graph inference results.

~ *Co-regulation a priori* ~ Gene regulation is not a simple causal process where one given TF acts on one given gene. Indeed, gene regulation *via* TFs calls in complex mechanisms involving several TFs which may act in cooperation. This cooperation can be viewed as a combinatorial regulation or as a co-regulation. Combinatorial regulation is observed when multiple TFs activate or repress the target gene in an independent manner. The co-regulation implies a dependence among TFs. They have to be associated to activate or to repress the target gene. Both mechanisms are illustrated in Figure 4.2.



(a) Combinatorial regulation. *(b) Co-regulation.*

Figure 4.2 **~** TFs COOPERATION MECHANISMS FOR GENE EXPRESSION REGULATION **~** *In (a),* TFs *activate the transcription of the target gene independently while in (b)* TFs *have to be associated before activating the target gene transcription.*

As mentioned in Section 2.4, detecting combinatorial regulation from gene expression profiles or gene-gene interaction scores is a difficult task. However, detecting genes that are co-regulated by a couple of TFs can be easier. We thus integrate a co-regulation *a priori* in the edge selection task. This *a priori* encodes the fact that if two TFs are identified as co-regulators of a given gene, we consider plausible that this couple of co-regulators can act similarly on the other genes.

How to identify co-regulation? We recall that we work with a fully-connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, weighted by gene-gene interaction scores $\omega_{i,j}$, obtained from gene expression profiles. Some of nodes in the graph are known to be putative TFs and play a central role. For a given couple of TFs (j, j'), regulation of a given $\overline{\text{TF}}$ k may be detected if the weight $\omega_{k,j}$ and $\omega_{k,j'}$ are both higher than an arbitrary threshold γ. This detection leads to the combinatorial regulation only. An association between TFs, required for co-regulation mechanisms (Figure 4.2(b)), is assumed whether the $\omega_{j,j'}$ is higher than γ. A couple (j, j') of TFs is thus identified as co-regulators if the following condition is verified:

$$\min\{\omega_{j,j'}, \omega_{k,j}, \omega_{k,j'}\} > \gamma. \tag{4.3}$$

For a given $\overline{\text{TF}}$ i and couple of TFs (j, j'), a stronger identification may result in counting the number of other $\overline{\text{TF}}$s which verify the Condition (4.3). Normalizing by the number of $\overline{\text{TF}}$s minus 1 allows us to define a probability of co-regulation, denoted by $\rho_{i,j,j'}$ and taking the following form:

$$\rho_{i,j,j'} = \mu \frac{\sum_{k \in \mathbb{V} \setminus (\mathbb{T} \cup \{i\})} \mathbb{1}\left(\min\{\omega_{j,j'}, \omega_{k,j}, \omega_{k,j'}\} > \gamma\right)}{|\mathcal{V} \setminus \mathcal{T}| - 1}, \tag{4.4}$$

where $\mu \geq 0$ is a parameter controlling the global impact of the *a priori* on the global cost while $\mathbb{1}(\cdot)$ is the characteristic function and equals 1 if its argument is verified and 0 otherwise. The probability of co-regulation is thus used to enforce co-regulation *a priori* if it is non-null.

How to integrate co-regulation a priori? From the probability of co-regulation, we are now able to decide to which TFs couples the co-regulation *a priori* have to be applied. We recall that if a couple of TFs is identified as co-regulators, the co-regulation *via* these two TFs is enforced for the $\overline{\text{TF}}$s. In other words, if (j, j') denotes a couple of co-regulators identified *via* $\rho_{i,j,j'}$, inference of $e_{i,j}$ and $e_{i,j'}$ is coupled. This co-regulation *a priori* is illustrated in Figure 4.3.

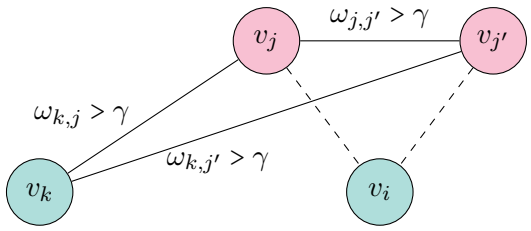


Figure 4.3 ~ CO-REGULATION *a priori* EFFECT ON EDGE SELECTION ~
If a couple of TFs (j, j') *is identified as co-regulators for a* $\overline{\text{TF}}$ k *(verified condition (4.3) represented by solid edges), the presence in the inferred graph of edge* $e_{i,j}$ *is coupled with the presence of* $e_{i,j'}$*, for all* $\overline{\text{TF}}$ i *different from* $\overline{\text{TF}}$ k.

Coupling the inference of edges $e_{i,j}$ and $e_{i,j'}$ is equivalent to enforcing corresponding labels $x_{i,j}$ and $x_{i,j'}$ to be the same. Moreover, the enforcement of the coupling should scale with the

strength of the probability $\rho_{i,j,j'}$. As result, we mathematically translate the proposed biological *a priori* is formulated as follows:

$$\psi(x_{i,j}, x_{i,j'}) = \sum_{\substack{i \in \mathbb{V} \setminus \mathbb{T} \\ (j,j') \in \mathbb{T}^2, j'>j}} \rho_{i,j,j'} \, |x_{i,j} - x_{i,j'}|, \tag{4.5}$$

Taking into account the two presented *a priori*, the edge selection problem can be expressed as the following energy minimization:

$$\underset{\boldsymbol{x} \in \{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j) \in \mathbb{V}^2 \\ j>i}} \omega_{i,j} \, |x_{i,j} - 1| \quad + \quad \lambda_{i,j} \, x_{i,j} \quad + \quad \sum_{\substack{i \in \mathbb{V} \setminus \mathbb{T} \\ (j,j') \in \mathbb{T}^2, j'>j}} \rho_{i,j,j'} \, |x_{i,j} - x_{i,j'}|. \tag{4.6}$$

The proposed functional to be minimized is compatible with energy functions minimized by Graph Cuts. We thus now detail how to use the Graph Cuts framework to solve (4.6).

4.1.2 Optimization *via* a maximal flow framework

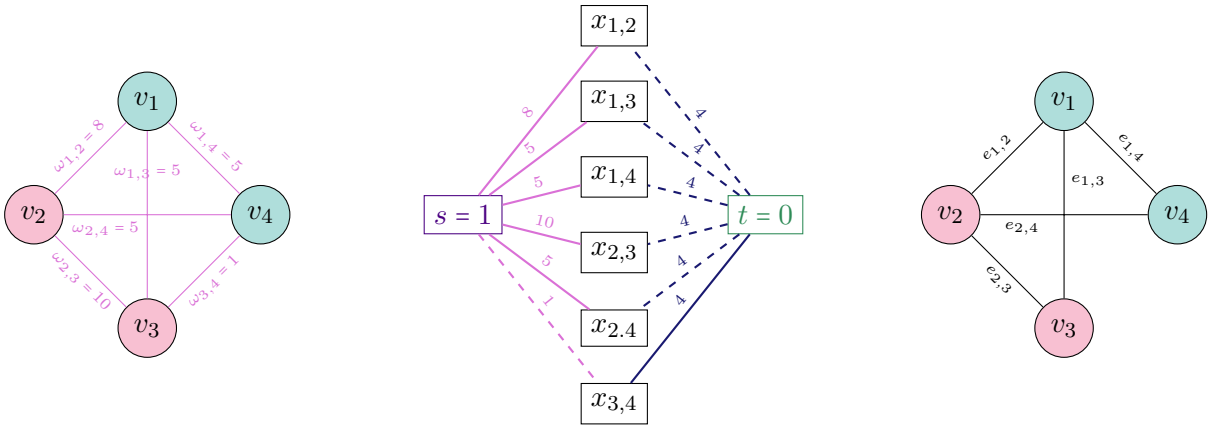
The minimum cut problem (4.6) allows us to compute an optimal edge labeling. As mentioned in Section 3.3.2, it can be solved by maximizing a flow in a transportation network. Kolmogorov and Zabih (2004) provide rules to construct a transportation network $\mathcal{G}_f$ corresponding to the underlying minimum cut problem.

~ *Flow network construction* ~ Before presenting the construction of $\mathcal{G}_f$ corresponding to Problem (4.6), we firstly expose the construction of a flow network corresponding to the classical thresholding (CT) problem. Even if trivial and without practical use, CT can be expressed in a Graph Cuts framework, where it shrinks to a particular case of (4.6):

$$\underset{\boldsymbol{x} \in \{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j) \in \mathbb{V}^2 \\ j>i}} \omega_{i,j} \, |x_{i,j} - 1| \quad + \quad \lambda \, |x_{i,j} - 0|. \tag{4.7}$$

In (4.7) variables $x_{i,j}$ correspond to the edge labeling. In the corresponding flow network, these variables are associated to nodes. A source node s, labeled by 1 and a sink node t, labeled by 0, are also added. Nodes coding for $x_{i,j}$ variables are linked to the source s by edges weighted by $\omega_{i,j}$ and to the sink t by edges weighted by λ. An illustration of the flow network construction for CT in a toy example is displayed in Figure 4.4.

Construction of the flow network for *BRANE Cut* (4.6) is based on the same principle but its construction is more complex due to the non-null probabilities of co-regulation $\rho_{i,j,j'}$ and the node-dependent $\lambda_{i,j}$ values. Firstly, probabilities of co-regulation are factors of $|x_{i,j} - x_{i,j'}|$. Thus, to take them into consideration in the flow network construction, an edge is added between nodes encoding edge variables $x_{i,j}$ and $x_{i,j'}$ if the probability $\rho_{i,j,j'}$ is non-null. These additional edges are simply weighted by the corresponding $\rho_{i,j,j'}$ weights. Secondly, we recall that weights $\lambda_{i,j}$ differ depending on the type of nodes i and j. Indeed, as described in Equation (4.2), $\lambda_{i,j}$ can depend on η, λ_{TF} or $\lambda_{\overline{\mathrm{TF}}}$. To take into account the plausible multiple values of weights $\lambda_{i,j}$,



(a) Initial complete graph. *(b) Flow network for CT (4.7).* *(c) Inferred graph with CT.*

Figure 4.4 ~ FLOW NETWORK CONSTRUCTION FOR CT PROBLEM ~
The initial graph (a) to be pruned is transformed into a transportation network (b) in which a maximal flow computation is performed to return an optimal edge labeling $\boldsymbol{x}^*$ leading to the inferred network (c). Pink nodes correspond to TF *nodes and green nodes to* $\overline{\text{TF}}$ *nodes. In (b), dashed edges correspond to saturated edges obtained after max-flow computing. We choose to present the case of unscaled weights and parameters i.e. $\omega_{i,j}$ and λ can take unbounded positive values. In this example, λ is set to 4.*

auxiliary nodes have to be added. They corresponds to gene nodes $\{v_1, \ldots, v_G\}$ of the complete graph $\mathcal{G}$ to prune. For each node $x_{i,j}$, two edges are added: from node $x_{i,j}$ to auxiliary nodes v_i with weight λ_i and from node $x_{i,j}$ to auxiliary nodes v_j with weight λ_j. Weights λ_i and λ_j are defined to be in accordance with $\lambda_{i,j} = \lambda_i + \lambda_j$. Values of weights λ_i and λ_j are provided in Table 4.1 according to the nature of nodes v_i and v_j. Finally, the node composition of the flow network for *BRANE Cut* is:

- one source s, labeled by 1,
- one sink t, labeled by 0,
- E nodes labeled by $x_{i,j}$. They encode edge binary variables to be optimized in $\mathcal{G}$ and take 0 or 1,
- G auxiliary nodes $\{v_1, \ldots, v_G\}$ to take into account the node-dependent weights $\lambda_{i,j}$ (second term of Equation (4.6)).

And the edge composition is:

- E edges between the node s and nodes $x_{i,j}$. Edge linking s to $x_{i,j}$ is weighted by $\omega_{i,j}$,
- $2E$ edges between nodes $x_{i,j}$ and the two corresponding nodes v_i and v_j. Edges linking $x_{i,j}$ to v_i and v_j are weighted by λ_i and λ_j, respectively. Their values are given in Table 4.1.

Nature of nodes	$\lambda_{i,j}$	λ_i	λ_j
$(v_i, v_j) \notin \mathcal{T}^2$	$2\,\eta$	η	η
$(v_i, v_j) \in \mathcal{T}^2$	$2\,\lambda_{\mathrm{TF}}$	λ_{TF}	λ_{TF}
$v_i \in \mathcal{T}$ and $v_j \notin \mathcal{T}$	$\lambda_{\mathrm{TF}} + \lambda_{\overline{\mathrm{TF}}}$	λ_{TF}	$\lambda_{\overline{\mathrm{TF}}}$
$v_i \notin \mathcal{T}$ and $v_j \in \mathcal{T}$	$\lambda_{\mathrm{TF}} + \lambda_{\overline{\mathrm{TF}}}$	$\lambda_{\overline{\mathrm{TF}}}$	λ_{TF}

Table 4.1 ~ Splitting scheme of the node-dependent $\lambda_{i,j}$ ~
The generic formulation of BRANE Cut involves a parameter $\lambda_{i,j}$ that takes different values according to the nature of the nodes i and j, TF or $\overline{\mathrm{TF}}$. The integration of this parameter in the transportation network of BRANE Cut require a splitting into two parameters λ_i and λ_j. Correspondence between values of λ_i, λ_j and $\lambda_{i,j}$ values is summed up in this table.

- q edges between nodes $x_{i,j}$ and $x_{i,j'}$ if the probability of co-regulation $\rho_{i,j,j'}$ is non-null. Edge linking $x_{i,j}$ to $x_{i,j'}$ is weighted by $\rho_{i,j,j'}$.
- G edges between node t and the nodes v_i, with infinite weights.

The structure of the transportation network $\mathcal{G}_f$ for BRANE Cut (4.6) for the previous toy example in Figure 4.4(a) is displayed in Figure 4.5.

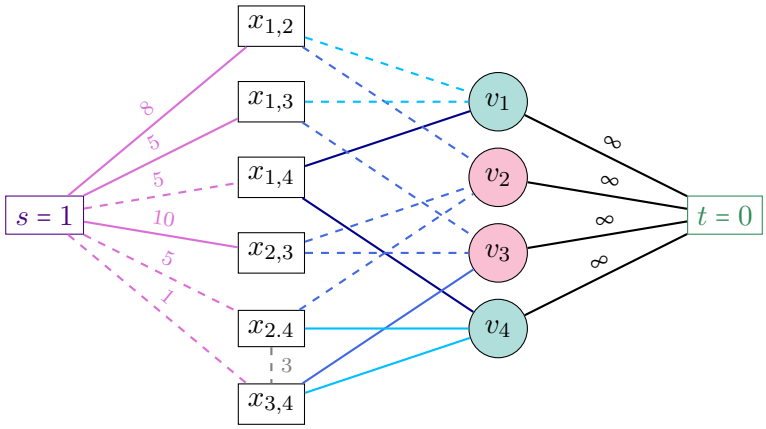


Figure 4.5 ~ Flow network construction for BRANE Cut ~
In this example, we fix $\eta = 6$, $\lambda_{\mathrm{TF}} = 3$ and $\lambda_{\overline{\mathrm{TF}}} = 1$. Taking $\gamma = 4$ implies that v_1, v_2 and v_3 satisfy the co-regulation a priori *leading to the presence of an additional edge between nodes $x_{2,4}$ and $x_{3,4}$, weighted by $\rho_{4,2,3} = 3$, when μ is set to 3. Maximum flow computation in such a graph leads to saturated edges (represented as dashed lines). The values from the source and the sink are propagated through non-saturated paths, thus leading to $x^*_{1.4} = x^*_{2,4} = x^*_{3,4} = 0$ and $x^*_{1,2} = x^*_{1,3} = x^*_{2,3} = 1$.*

Computing a maximal flow from the source to the sink in such a flow network saturates some edges, thus splitting nodes labeled by $x_{i,j}$ into two different groups: nodes that are reachable

through a non-saturated path from the source, and those that are not. Assuming that the source node s is labeled by 1 and the sink node t is labeled by 0, binary values are thus attributed to edge labels $x_{i,j}$ (secondarily, nodes v_i in the flow network are labeled by 0, without impacting optimal labeling computation for $x_{i,j}$). The final labeling on $x_{i,j}$ returns the set of selected edges $\mathcal{E}^*$ which minimizes (4.6).

~ *Problem dimension reduction* ~ As explained above, the optimal solution to the minimization problem (4.6) may be obtained *via* maximal flow computation in a network generated from the whole original graph $\mathcal{G}$. In practice, many co-regulation probabilities have zero values. Rather than building 0-valued edges in the flow network $\mathcal{G}_f$, reducing the dimension of this network is judicious. Indeed, if $\rho_{i,j,j'}$ is null, no link exists between node $x_{i,j}$ and $x_{i,j'}$ in the flow network $\mathcal{G}_f$. As a result, nodes $x_{i,j}$ and $x_{i,j'}$ are linked only to the source s and their auxiliary nodes (v_i, v_j) and (v_i, v'_j), respectively. In such a case, the optimal solution for $x_{i,j}$ and $x_{i,j'}$ takes an explicit form and can be computed without constructing the flow network $\mathcal{G}_f$. More generally, for all $x_{i,k}$, $i \in \mathbb{V}$, $k \in \{j, j'\}$, $(j, j') \in \mathbb{T}^2, j' > j$ such that $\rho_{i,j,j'} = 0$, the optimal solution is trivial:

$$x^*_{i,k} = \begin{cases} 1 & \text{if } \omega_{i,k} > \lambda_{i,k}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

On the contrary, if $\rho_{i,j,j'}$ is non-null, a link is present between nodes $x_{i,j}$ and $x_{i,j'}$. In such a case, non trivial solutions exist and these nodes have to be taken into account in the flow network $\mathcal{G}_f$. Finally, the flow network is constructed only for all $x_{i,k}$, $i \in \mathbb{V}$, $k \in \{j, j'\}$, $(j, j') \in \mathbb{T}^2, j' > j$ such that $\rho_{i,j,j'} \neq 0$. The resulting node composition for $\mathcal{G}_f$ is: a source s, the edge labeling nodes $x_{i,k}$ verifying the above condition, the corresponding gene nodes v_i and v_k and the sink t.

In the toy example in Figure 4.5, only nodes $x_{2,4}$ and $x_{3,4}$ are linked together. Thus, Equation (4.8) is used to compute the optimal labeling of nodes $x_{1,2}$, $x_{1,3}$, $x_{1,4}$, $x_{2,3}$. The flow network is constructed taking into account nodes $x_{2,4}$ and $x_{3,4}$ only (as well as their respective auxiliary nodes v), as illustrated in Figure 4.6.

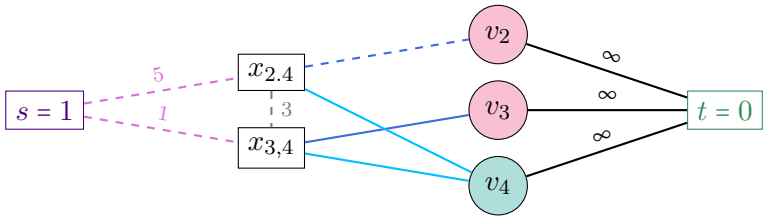


Figure 4.6 **~** FLOW NETWORK CONSTRUCTION FOR *BRANE Cut* AFTER DIMENSION REDUCTION **~**
The dimension reduction is obtained by keeping all nodes $x_{i,j}$ and $x_{i,j'}$ for which the co-regulation probability $\rho_{i,j,j'}$ is non-null. Only the involving auxiliary nodes v_i and v_j are preserved.

This reduction dimension trick dramatically decreases the size of the flow network $\mathcal{G}_f$, leading to a fast optimization strategy to generate a solution to the proposed variational formulation

(4.6). One the advantages of employing the *BRANE Cut* algorithm is the optimality guarantee of the resulting inferred network with respect to the proposed criterion.

4.1.3 Objective results and biological interpretation

As detailed in Section 3.2, we assess *BRANE Cut* performances, in terms of Area Under the Precision-Recall curve (AUPR), on datasets provided by the DREAM4 and DREAM5 challenges. From each dataset, a weighted complete graph is firstly computed using one of the two following state-of-the-art GRN inference methods: CLR (Faith *et al.*, 2007) and GENIE3 (Huynh-Thu *et al.*, 2010). Then, edge selections parametrized by λs and yielding final GRNs to be evaluated, are performed by both the classical thresholding (CT) and our *BRANE Cut* approach. AUPR for CT and *BRANE Cut* on CLR and GENIE3 weights are then computed and compared. We also used the post-processing Network Deconvolution (Feizi *et al.*, 2013) on CLR and GENIE3 weights. This step provides a novel set of weights, respectively denoted by ND-CLR and ND-GENIE3. CT and *BRANE Cut* are also applied on these novel weights for additional comparisons. Note that all our *BRANE Cut* simulations are performed with the same data-driven parameters setting for which details are provided further away in a dedicated part (Section 4.1.3 - p. 95).

~ *A close-up on AUPR curves* **~** Before detailing numerical results, let us discuss about Precision-Recall (PR) curve comparison with AUPRs in a GRN context. A qualitative interpretation can be made from PR curves, by comparing their relative location — above means better.

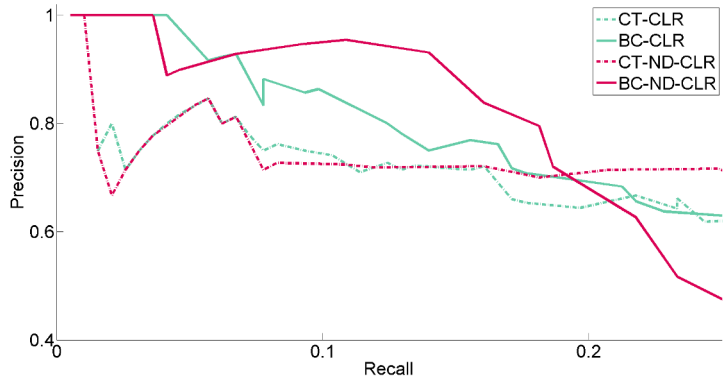
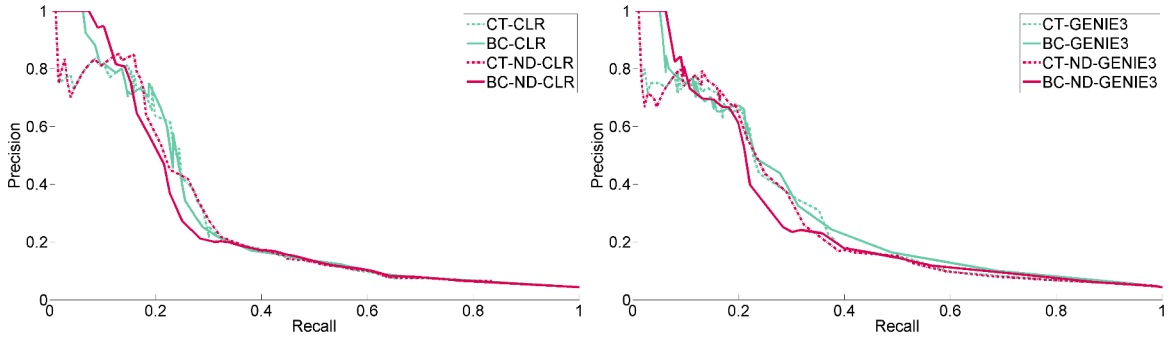


Figure 4.7 **~** Zoom on the top-left part of a PR curve **~**

For instance in Figure 4.7, without entering here in simulation details, we observe that, on the top-left, both solid lines (green and pink) are above dashed lines (green and pink). In other words, *BRANE Cut* offers higher performance than CT using either CLR (green) or ND-CLR (pink) weights. However, this relative order may change for higher Recall values (crossing around a Recall of 0.2). Due to these potential crossings, quantitative assessment is traditionally preferred through the Area Under the Precision-Recall curve (AUPR) as detailed in Section 3.2.2. Computed on whole PR curves, this measure provides a global quantitative performance across the whole range of thresholds λ.

Notwithstanding, not all inferred GRNs are of our interest. With a Precision lower than 50 %, less than half of the selected edges are genuine. Resulting networks are thus biologically untrustworthy and suffer from a poor predictive power. This matters is important as GRNs are employed to provide insight for costly biological experiments. Hence, biologically interpretable networks are found for high Precision and, unfortunately, low Recall values i.e. on the top-left part of PR curves. It is thus interesting to also emphasize the performance on this part in addition to the global AUPR. Results become more pertinent if both global and local improvements are observed. Notably, high-precision improvement can counterbalance unfavorable crossing of PR curves in areas of lesser biological importance. Keeping in mind these subtleties inherent to GRNs, we now proceed to numerical and biological results.

~ *Numerical results on simulated datasets* ~ *BRANE Cut* performance is firstly assessed on the five datasets provided by the DREAM4 challenge (Marbach *et al.*, 2010). PR curves obtained with CT and *BRANE Cut* on CLR, ND-CLR, GENIE3 and ND-GENIE3 weights for the five simulated datasets are provided in Figures 4.8 to 4.12.

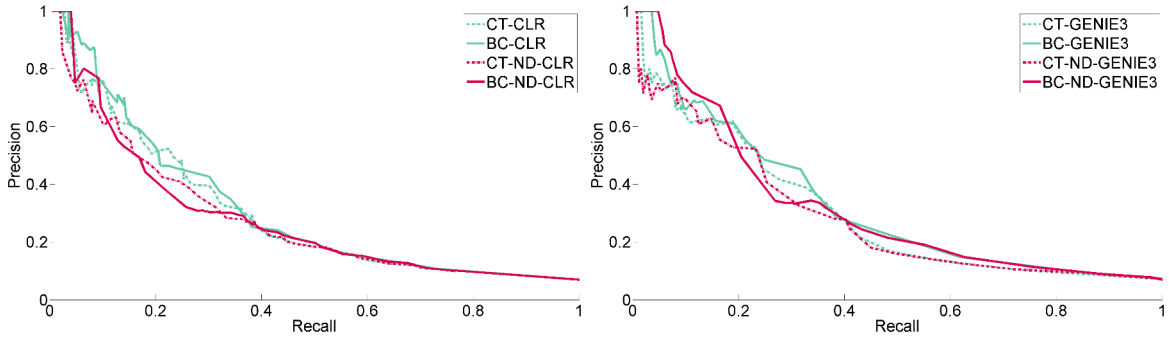


(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 4.8 ~ PR CURVES FOR THE DATASET 1 OF DREAM4 (*BRANE Cut*) ~
Precision-Recall (PR) curves obtained using CT or BRANE Cut on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.

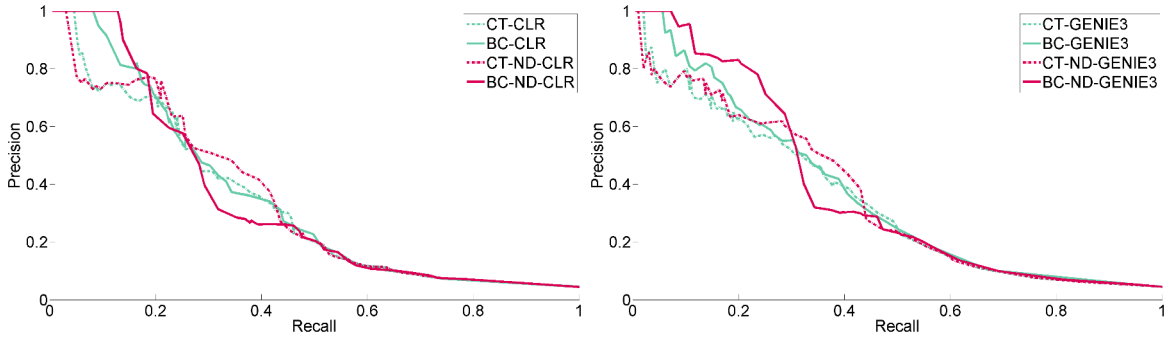
The associated AUPRs (Area Under PR curves) are reported in Table 4.2(a). They highlight in italics that, globally, first and second best performances are always produced with *BRANE Cut*. Furthermore, each method tested (CLR, GENIE3, ND-CLR or ND-GENIE3) used as initialization exhibits an improved AUPR with *BRANE Cut* post-processing. Indeed, the average improvement reaches 10.9 % based on the CLR weights, 8.4 % for the GENIE3 weights, 5.9 % with ND-CLR weights and 7.2 % compared to the ND-GENIE3 weights, see Table 4.2(b). In other words, using *BRANE Cut* is always beneficial to these datasets.

We recall here that ND (Feizi *et al.*, 2013) is a post-processing method. Hence, in addition to comparing CT and *BRANE Cut* on CLR and GENIE3 weights and their respective improved weights by ND, we can also assess the post-processing itself. For this purpose, performances of CT on ND-CLR or ND-GENIE3 are compared to those obtained with *BRANE Cut* on CLR and



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 4.9 ~ PR CURVES FOR THE DATASET 2 OF DREAM4 (*BRANE Cut*) ~
Precision-Recall (PR) curves obtained using CT or BRANE Cut on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.

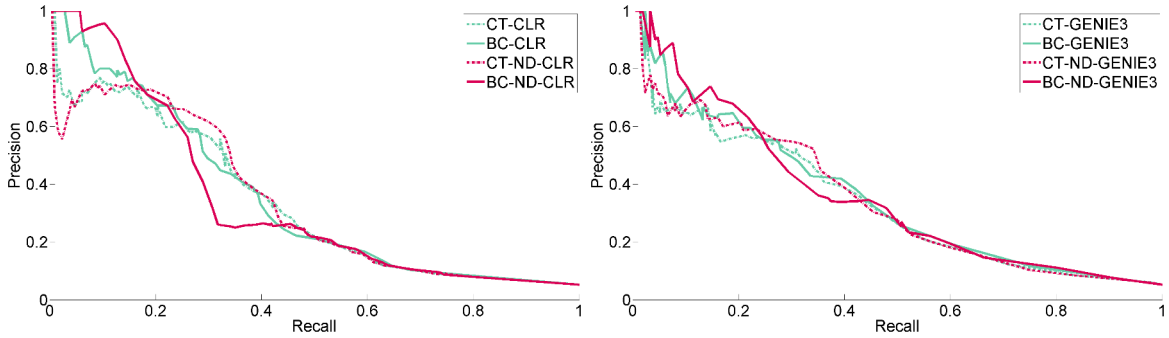


(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 4.10 ~ PR CURVES FOR THE DATASET 3 OF DREAM4 (*BRANE Cut*) ~
Precision-Recall (PR) curves obtained using CT or BRANE Cut on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.

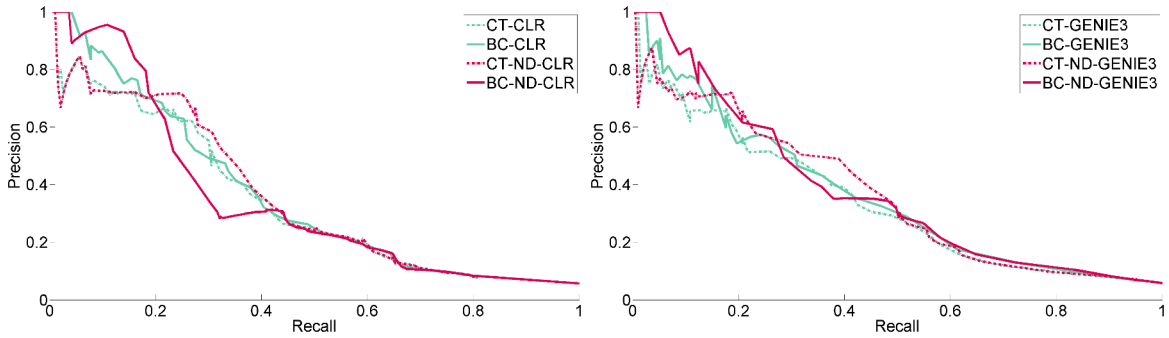
GENIE3. As shown in Table 4.3, *BRANE Cut* outperforms Network Deconvolution except for a practically unnoticeable degradation on the fifth network for GENIE3 weights. Nevertheless, the degradation we observe is essentially located in areas of lesser biological importance and high-precision performance are noticeable with an improvement ratio of 1.28 in the Precision range of [80-100].

On the Precision-Recall curves in Figures 4.8 to 4.12, we notice that the improvements of our results are mostly obtained in the first part of the curves, generally corresponding to a Precision greater than 50 % in the inference. Thus, such inferred graphs are expected to be more reliable for a biological interpretation. From this observation, looking at the AUPR for different Precision ranges — from the whole scale to precisions above 90 % — provides a finer assessment of the predictive power of inference methods. Thus, Figure 4.13 highlights relative



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 4.11 **~** PR CURVES FOR THE DATASET 4 OF DREAM4 (*BRANE Cut*) **~**
Precision-Recall (PR) curves obtained using CT or BRANE Cut on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 4.12 **~** PR CURVES FOR THE DATASET 5 OF DREAM4 (*BRANE Cut*) **~**
Precision-Recall (PR) curves obtained using CT or BRANE Cut on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.

AUPR improvements, for given Precision ranges, obtained for the five datasets of the DREAM4 multifactorial challenge and the four weight sets: CLR, ND-CLR, GENIE3 and ND-GENIE3. Figure 4.13 illustrates that *BRANE Cut* improvement ratios over AUPR — from various weights — are clearly visible at higher Precision ranges, typically over 65 %. Improvement ratios refer to the ratio between the AUPR of *BRANE Cut* and CT in the selected Precision range. They allow to evaluate *BRANE Cut* performance on specific areas instead of assessing the global performance. This procedure makes sense as we recall that biologically interpretable networks are found in a restricted area of the PR curve, notably for high precision and low recall. For instance, on Network 2, computing AUPR on the upper Precision range from 80 to 100, *BRANE Cut* yields more significant improvement ratios of 2.7, 1.1, 4.4, 9.6 (with CLR, ND-CLR, GENIE3 and ND-GENIE3 weights, respectively). The improvement even becomes severalfold for the upmost Precision ranges. Based on the above global and range-based AUPR criteria, we conclude that

Dataset	1	2	3	4	5	Average
CT-CLR	0.256	0.275	0.314	0.313	0.313	0.294
BC-CLR	*0.282*	0.308	0.343	*0.344*	*0.356*	0.327
CT-GENIE3	0.269	0.288	0.331	0.323	0.329	0.308
BC-GENIE3	*0.298*	*0.316*	*0.357*	*0.344*	0.352	*0.333*
CT-ND-CLR	0.254	0.250	0.324	0.318	0.331	0.295
BC-ND-CLR	0.271	0.277	0.334	0.335	0.343	0.312
CT-ND-GENIE3	0.263	0.275	0.336	0.328	0.354	0.309
BC-ND-GENIE3	0.275	*0.312*	*0.367*	*0.346*	*0.368*	*0.334*

(a) AUPRs.

Dataset	1	2	3	4	5	Average
BC-CLR *vs* CT-CLR	10.1 %	11.8 %	9.1 %	9.9 %	13.7 %	10.9 %
BC-GENIE3 *vs* CT-GENIE3	10.7 %	9.9 %	7.8 %	6.5 %	7.0 %	8.4 %
BC-ND-CLR *vs* CT-ND-CLR	6.6 %	10.7 %	3.0 %	5.5 %	3.7 %	5.9 %
BC-ND-GENIE3 *vs* CT-ND-GENIE3	4.4 %	13.4 %	9.2 %	5.4 %	3.8 %	7.2 %

(b) Relative gains.

Table 4.2 **~** NUMERICAL PERFORMANCE ON DREAM4 (*BRANE Cut*) **~**
(a) Area Under PR curve (AUPR) obtained using CT or BRANE Cut (BC) on CLR, ND-CLR, GENIE3 and ND-GENIE3 weights. Weights are computed for each dataset (1 to 5) of the DREAM4 multifactorial challenge. Average AUPR are also reported as well as the two maximal improvements (in italic). (b) Relative gains obtained by comparing BRANE Cut to CT.

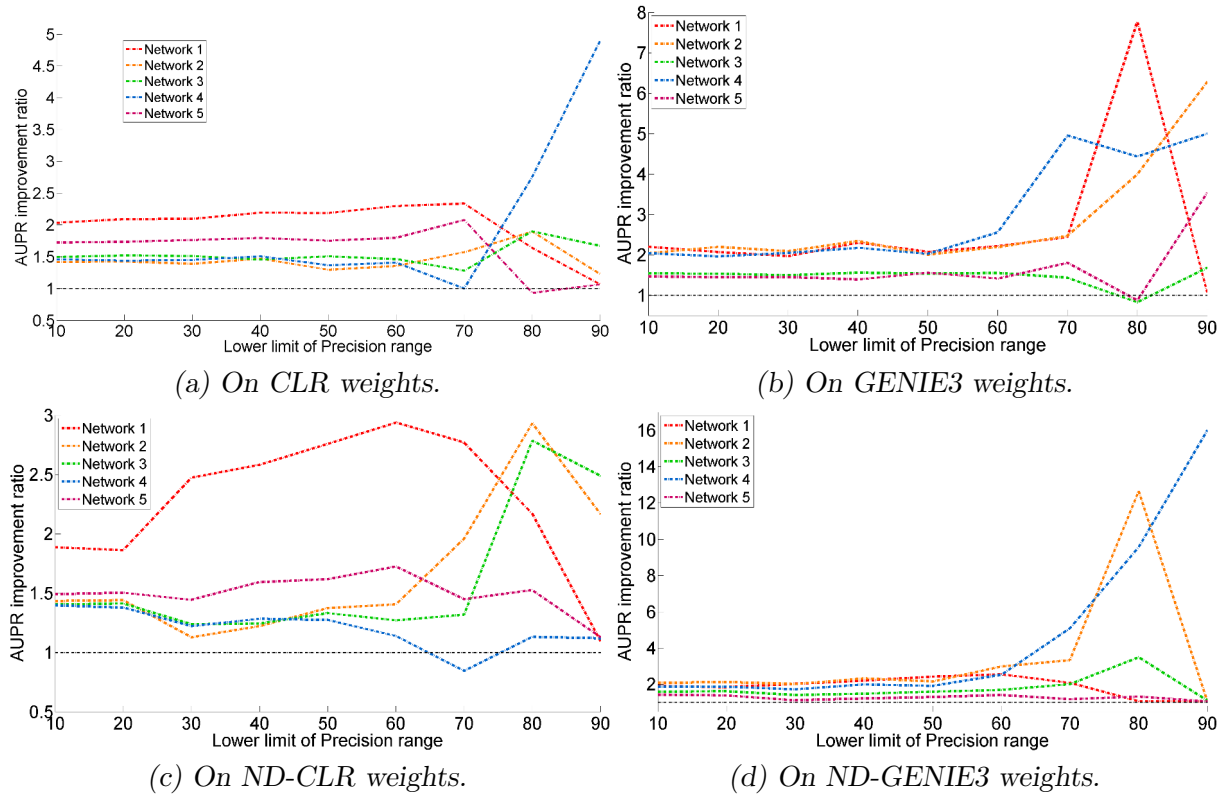
BRANE Cut outperforms state-of-the-art methods on the simulated datasets provided by the DREAM4 multifactorial challenge. Specifically, classical thresholding (CT) results are sensibly refined by our approach, regardless of initial weights and post-processing. In other words, the use of *BRANE Cut* can be considered as most probably beneficial for inference.

From the positive objective results obtained on the simulated datasets from the DREAM4 multifactorial challenge, *BRANE Cut* is also assessed on a more realistic simulated dataset provided by the DREAM5 challenge, see Section 3.2.1. Precision-Recall curves are displayed in Figure 4.14 and associated AUPRs and relative gains are provided in Table 4.4. As for previous results, *BRANE Cut* shows refined results compared to classical thresholding (CT), with a maximal improvement reaching about 6 %.

In view of the positive results, we assess *BRANE Cut* on real transcriptomic data from the bacteria *Escherichia coli*. We present both numerical results and the biological interpretation extracted from an inferred network by *BRANE Cut* on GENIE3 weights.

Dataset	1	2	3	4	5	Average
BC-CLR *vs* CT-ND-CLR	11 %	23.2 %	5.9 %	8.2 %	7.5 %	11.2 %
BC-GENIE3 *vs* CT-ND-GENIE3	13.8 %	14.9 %	6.2 %	4.9 %	−0.6 %	7.7 %

Table 4.3 ~ POST-PROCESSING PERFORMANCE ON DREAM4 (*BRANE Cut*) ~
Relative gains computed using AUPRs provided in Table 4.2(a) and are given for BRANE Cut using CLR (resp. GENIE3) weights compared to CT using ND-CLR (resp. ND-GENIE3).

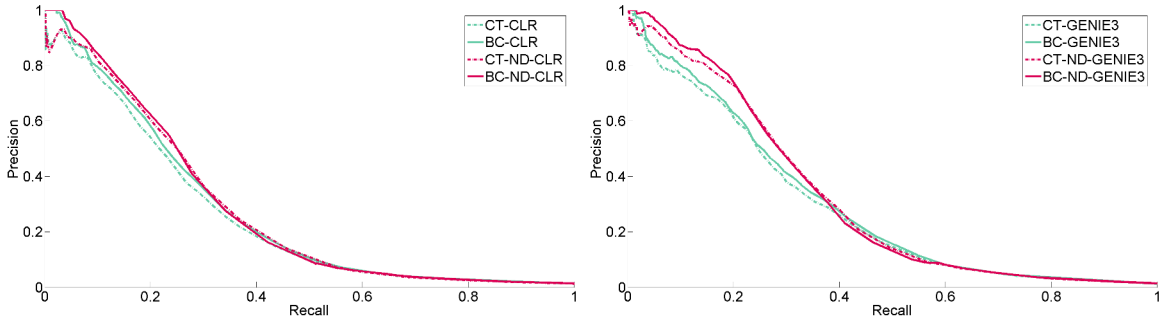


(a) On CLR weights. *(b) On GENIE3 weights.*

(c) On ND-CLR weights. *(d) On ND-GENIE3 weights.*

Figure 4.13 ~ RANGE-PRECISION-DEPENDENT PERFORMANCE ON DREAM4 ~
Differential improvement over the Precision are shown through relative AUPR, computed for PR curves in Figures 4.8 to 4.12 at different selected Precision ranges: [10,100], [20,100], ..., [90,100]. Here, the improvement is defined as the AUPR ratio of BRANE Cut and CT on (a) CLR, (b) GENIE3, (c) ND-CLR and (d) ND-GENIE3 weights.

~ *Numerical performance on the Escherichia coli dataset* ~ CLR and GENIE3 weights are firstly computed from the *E. coli* dataset presented in Section 3.2.1. Network Deconvolution post-processing is then applied on both CLR and GENIE3 weights yielding ND-CLR and ND-GENIE3 weights. As previously, for a given set of weights, varying λ values for both CT and *BRANE Cut* allows us to draw the Precision-Recall curves displayed in Figure 4.15. Corresponding AUPRs and relative gains obtained by *BRANE Cut* against CT are provided in Table 4.5.



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

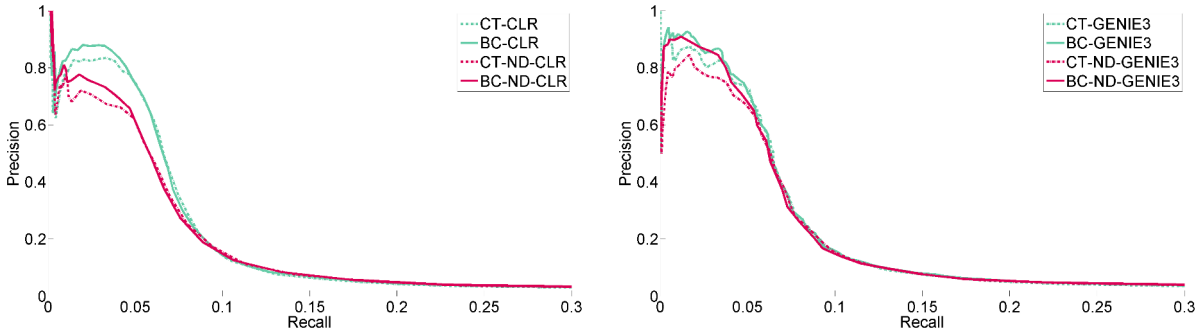
Figure 4.14 ~ PR CURVES FOR THE DATASET 1 OF DREAM5 (*BRANE Cut*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Cut on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

	AUPR	Gain		AUPR	Gain
CT-CLR	0.252	6.3 %	CT-ND-CLR	0.272	1.9 %
BC-CLR	0.268		BC-ND-CLR	0.277	
CT-GENIE3	0.283	4.2 %	CT-ND-GENIE3	0.313	1.1 %
BC-GENIE3	*0.295*		BC-ND-GENIE3	*0.317*	

Table 4.4 ~ NUMERICAL PERFORMANCE ON THE DATASET 1 OF DREAM5 (*BRANE Cut*) ~ *Area Under Precision-Recall curve (AUPR) obtained using CT or BRANE Cut on CLR, ND-CLR, GENIE3 or ND-GENIE3 weights computed from dataset 1 of the DREAM5 challenge. Relative gains between CT and BRANE Cut are also reported.*

Before discussing about comparative results and *BRANE Cut* performance, it is interesting to note the degraded behavior of all inference methods on real data. Indeed, while inference methods are able — on simulated data — to reach up to the third of the ground truth behavior, all performances decrease to less than one tenth. This particularity results in the fact that inferred networks promptly become inaccurate, especially due to the large amount of genes compared to the number of observations. We observe in this dataset, that networks with a precision greater than 60 % (and thus assumed accurate) correspond to small networks with less than 300 edges. Due to their higher predictive power and their readability, such small networks are often preferred by biologists and efforts have to be engaged to improve them particularly.

BRANE Cut performance obtained on real data strengthens results obtained on simulated data with a maximal improvement reaching 11.6 % with respect to a single-thresholding. As expected and previously observed, improvements concern the upper left side of Precision-Recall curves. This observation is illustrated in Figure 4.16, where a finer assessment is performed through various Precision ranges. Prominent improvements are thus observed for a Precision



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 4.15 ~ PR CURVES FOR THE *Escherichia coli* DATASET (*BRANE Cut*) ~ *Precision-Recall (PR) curves obtained using CT or BRANECut on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

	AUPR	Gain		AUPR	Gain
CT-CLR	0.0786	11.2 %	CT-ND-CLR	0.0715	11.6 %
BC-CLR	0.0874		BC-ND-CLR	0.0798	
CT-GENIE3	0.0890	3.0 %	CT-ND-GENIE3	0.0864	3.7 %
BC-GENIE3	*0.0917*		BC-ND-GENIE3	*0.0896*	

Table 4.5 ~ NUMERICAL PERFORMANCE ON *Escherichia coli* DATASET (*BRANE Cut*) ~ *Area Under Precision-Recall curve (AUPR) obtained using CT or BRANE Cut on CLR, ND-CLR, GENIE3 or ND-GENIE3 weights computed from the* Escherichia coli *dataset. Relative gain between CT and BRANE Cut are also reported.*

greater than 65 %. Finding such promising numerical results on this real data encourages us to assess the biological relevance of the inferred networks.

~ Biological validation ~ Biological relevance is assessed through the added information gain on a network inferred by *BRANE Cut* compared to the one obtained with CT using the same initial weights. These weights are computed from the *E. coli* dataset with the GENIE3 method. Then, we select the *BRANE Cut* network with a Precision score of 85 %, corresponding to the best compromise in size and improvement. Network characteristics — in terms of Precision, Recall, number of TP and FP edges in common or specific to CT and *BRANE Cut* — are summarized in Figure 4.17.

When we compare the networks obtained with *BRANE Cut* and CT, we observe that for the same Precision score, *BRANE Cut* is able to generate a larger graph than CT, with 54 additional edges. Putting common edges aside, we remark that *BRANE Cut* specifically infer 48 true edges while CT specifically infer 4 true links only. In addition to comparing positive results, it is interesting to evaluate the biological relevance of potential wrongly inferred edges (or predic-

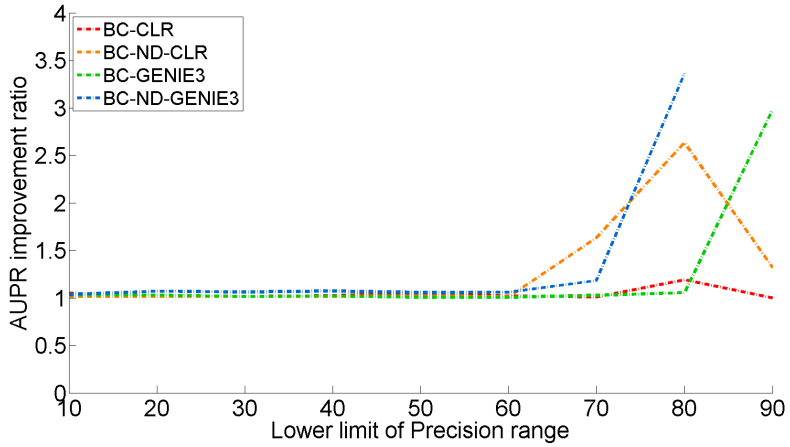


Figure 4.16 ~ RANGE-PRECISION-DEPENDENT PERFORMANCE ON *Escherichia coli* DATASET ~
Differential improvement over the Precision are shown through relative AUPR, computed for PR curves in Figure 4.15 at different selected range of Precision: [10,100], [20,100], ..., [90,100]. Here, the improvement is defined as the AUPR ratio of BRANE Cut and CT on (a) CLR, (b) GENIE3, (c) ND-CLR and (d) ND-GENIE3 weights.

tions) — 12 with *BRANE Cut* and 2 with CT. Predictions specifically obtained by *BRANE Cut* are displayed as solid green edges in the inferred *E. coli* network of the Figure 4.18.

As mentioned in Section 3.2.2, biological relevance of predictions are assessed through various databases such as RegulonDB (Gama-Castro *et al.*, 2016), EcoCyc (Keseler *et al.*, 2013) or STRING (Franceschini *et al.*, 2013). Among the 12 studied predictions — *flhC-flgK*, *flhC-fliD*, *flhD-cheA*, *fecI-cirA*, *fecI-entE*, *fecI-exbB*, *fecI-ybdB*, *lrp-argI*, *lrp-dppA*, *nac-glnK*, *nac-amtB* and *yhiE-yhiD* — 6 are recovered as direct links in the STRING database for which details are reported in Table 4.6. Among the 6 remaining predictions, 4 can be validated in terms of co-expression effect more than regulatory effects: *flhC-fliD*, *fecI-cirA*, *fecI-entE*, and *fecI-ybdB*. Even if all regulatory links are not validated as such, 10 predictions among the 12 make sense and seem to be biologically relevant. Figure 4.19 summarizes the biological assessment of the *BRANE Cut* predictions.

~ *Parameter settings* ~ Our model (4.6) involves four parameters to be fixed: λ_{TF}, $\lambda_{\overline{\mathrm{TF}}}$, μ and γ. Let us focus on the two threshold parameters λ_{TF} and $\lambda_{\overline{\mathrm{TF}}}$. As explained in Section 4.1.1, our TF-connectivity prior make sense for $\lambda_{\mathrm{TF}} \geq \lambda_{\overline{\mathrm{TF}}}$. A simple linear dependence $\lambda_{\mathrm{TF}} = \beta\lambda_{\overline{\mathrm{TF}}}$, with $\beta \geq 1$ suffices to define a generalized inference formulation encompassing the classical formulation (CT) when $\beta = 1$. We fixed here β as a parameter based on the gene/TF cardinal ratio: $\beta = \frac{|\mathcal{V}|}{|\mathcal{T}|}$. This choice is consistent when no *a priori* is formulated on the TFs (i.e. all genes are considered as putative TFs). Hence, $\beta = 1$ and $\lambda_{\mathrm{TF}} = \lambda_{\overline{\mathrm{TF}}}$. In such a case, without knowledge on TFs, we recover CT for gene network. The $\lambda_{i,j}$ parameter now only depends on a single free

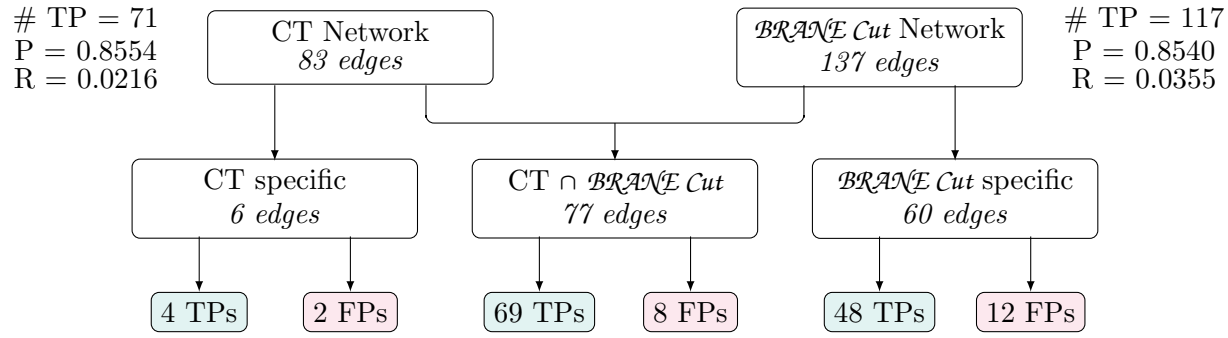


Figure 4.17 ~ CT AND *BRANE Cut Escherichia coli* NETWORK CHARACTERISTICS ~ *Networks are generated with CT or BRANE Cut on pre-computed GENIE3 weights from the* E. coli *dataset.*

Prediction	Co-O	Co-E	Co-M	N	CS
flhC-flgL	0.417	0.226	0.068	-	0.542
nac-glnK	-	0.885	-	-	0.885
fecI-exbB	-	0.697	0.632	0.067	0.890
nac-amtB	-	0.895	-	-	0.895
flhD-cheA	-	0.426	0.639	0.557	0.907
yhiE-yhiD	-	0.785	0.652	-	0.921

Table 4.6 ~ SIGNIFICANT STRING SCORES FOR *BRANE Cut* PREDICTIONS ~ *STRING scores evaluate functional links between two genes and involve here probabilities based on co-occurrence across genomes (Co-O), co-expression (Co-E), co-mentioned in PubMed abstracts (Co-M), neighborhood in the genome (N). Combine Score (CS) is the final score taking account all the probabilities.*

parameter $\lambda_{\overline{\text{TF}}}$ (or λ_{TF}), similarly to the large majority of inference methods requiring a final thresholding step on their weights. Using this parameter setting for $\lambda_{i,j}$, the construction of the Precision-Recall curves is carried out by linearly varying $\lambda_{i,j}$ between 0 and 1. For this purpose, we choose to vary $\lambda_{\overline{\text{TF}}}$ linearly between 0 and $1/(1+\beta)$.

The $\gamma \in [0,1]$ parameter in (4.4) drives the probability of co-regulation. It is employed as a threshold to determine which couples of TFs can be assimilated to co-regulators. We define γ from robust statistics (Huber and Ronchetti, 2009) as the $(G-1)^{\text{th}}$ quantile of the weights. This heuristic was experimentally found after looking for the best γ parameter with both a greedy search and *via* a simplex algorithm (Nelder and Mead, 1965).

The μ parameter controls the impact of the co-regulation *a priori* in the global inference. Weights $\omega_{i,j}$ are employed to compute co-regulation probabilities $\rho_{i,j,j'}$. Different weight distributions lead to different sets of non-zero co-regulation probabilities. Consequently, they impact the optimal choice for μ. This is observed in the different μ values chosen for the tested net-

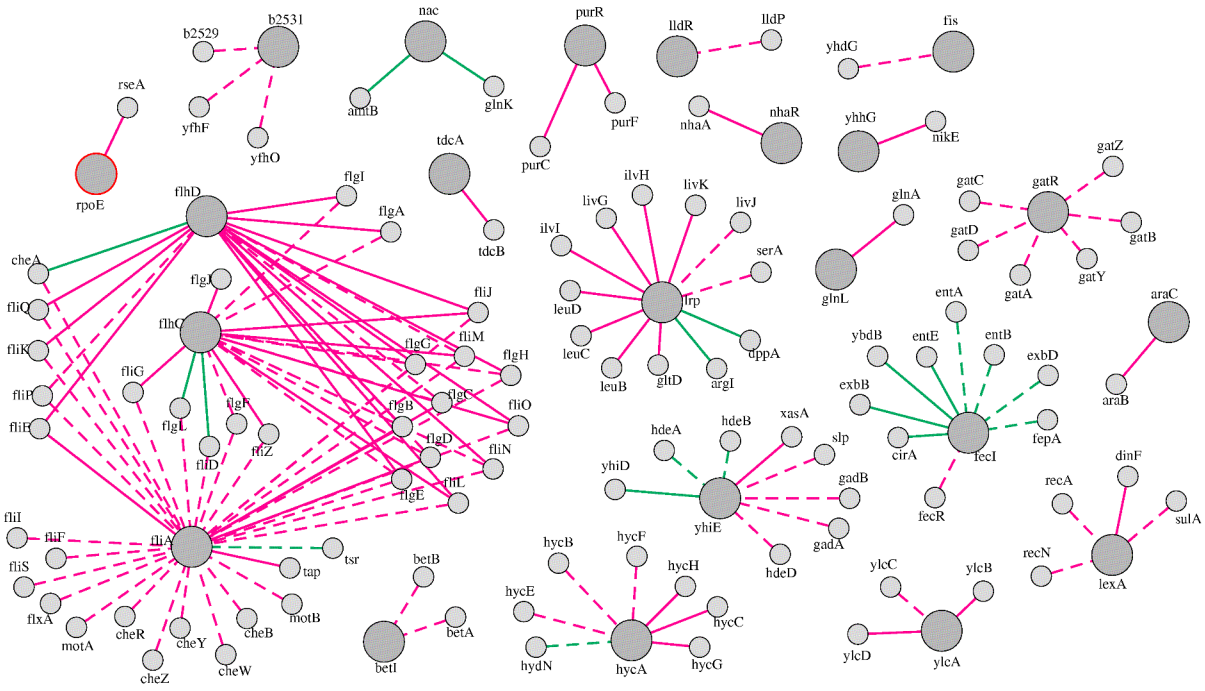


Figure 4.18 **~** INFERRED *Escherichia coli* NETWORK WITH *BRANE Cut* **~**
Network built using BRANE Cut using GENIE3 weights and containing 137 edges. Large dark gray nodes refers to TFs. *Inferred edges also reported in the ground truth are colored in pink while predictive edges are green. Dashed edges correspond to a link inferred by both BRANE Cut and GENIE3 while solid links refer to edges specifically inferred by BRANE Cut.*

works. For practically useful inference, we consider important to obtain a simple estimation of μ for a given network. It should also be of low sensitivity. For a given set of weights, we denote by C_r the number of identified couples of genes $(j, j') \in \mathbb{T}^2$ co-regulating at least one gene. The total number of co-regulator couples is equal to $\frac{|\mathcal{T}|(|\mathcal{T}|-1)}{2}$. We experimentally observe that an accurate order of magnitude close to the optimal μ is given by the cardinality-based ratio:

$$\mu = \frac{|\mathcal{T}|(|\mathcal{T}|-1)}{2C_r} \tag{4.9}$$

This heuristic is consistent with the biological view point, where a small proportion of co-regulator couples is expected.

In order to validate the two proposed data-driven heuristics, we assess results obtained with them in view of a sensitivity analysis of both μ and γ. The latter was performed on the five datasets of the DREAM4 challenge and using two kinds of initial weights (CLR and GENIE3). We vary the γ parameter with a step of 0.1 between 0.1 and its critical value for which no co-regulation is identified. For each γ value, the μ parameter is exhaustively assessed by varying it between 10 and 450 with a step equals 10. Results of the sensitivity analysis is illustrated

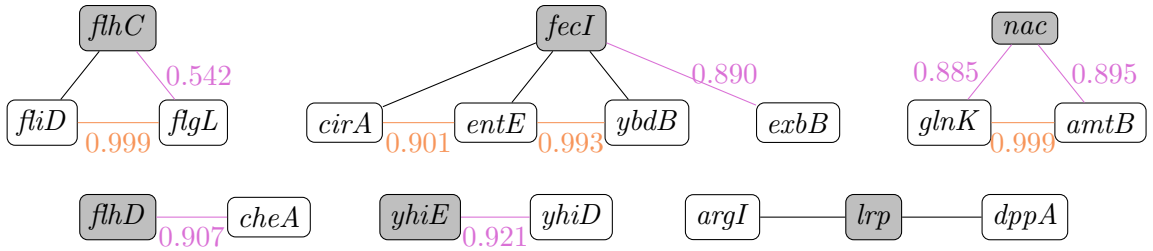
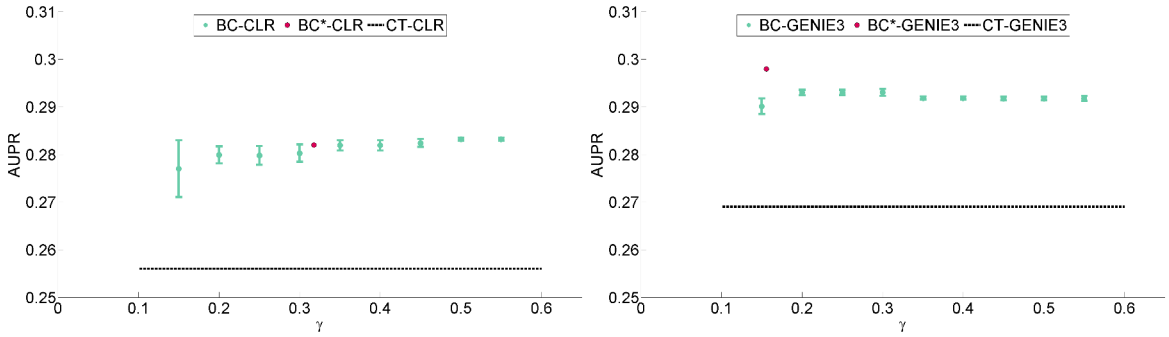


Figure 4.19 ~ *BRANE Cut* PREDICTIONS AND STRING VALIDATION ~
All links specifically inferred by BRANE Cut are reported and significant CS scores obtained with STRING. Purple scores and edges refer to direct link found in STRING database while orange scores and edges refer to direct link between targets.

in Figures 4.20 to 4.24. For each dataset and weight, we summarize the sensitivity analysis by averaging — for a given γ value — resulting AUPRs obtained by varying the μ parameter. The dispersion resulting in the choice of the μ is encoded through error bars.

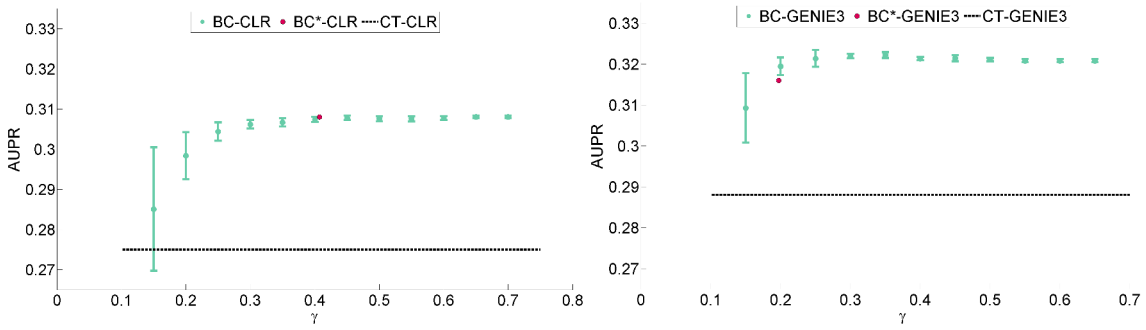


(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 4.20 ~ SENSITIVITY ANALYSIS OF μ AND γ ON THE DATASET 1 OF DREAM4 ~
Assessment of the parameter effects on AUPRs obtained using BRANE Cut on (a) CLR and (b) GENIE3 weights. For each γ, results obtained with BRANE Cut are given in terms of average AUPR and standard deviation over μ. BC-CLR refers to the AUPR results obtained with BRANE Cut parametrized by the data-driven heuristic. The AUPR obtained with CT as also recalled.*

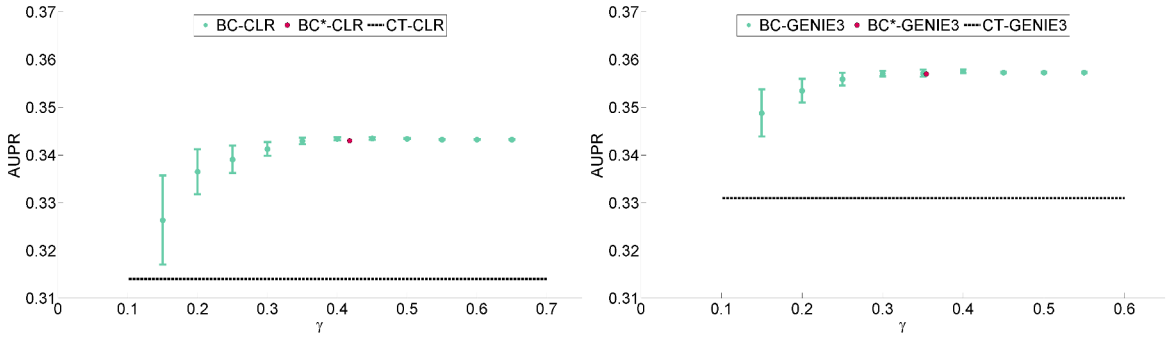
We first observe that, except for extremal parameter settings, *BRANE Cut* always outperforms the classical thresholding (CT). A low value of the γ parameter tends to decrease performance. This observation can be explained by the fact that a low γ value enforces a non-realistic number of co-regulation. In such a case, the value of the μ parameter yields dispersed AUPR as observed through the relatively large error bar. Using intermediate γ values, AUPR results appear stable over the μ parameter. Note that no co-regulation *a priori* is involved for high γ values.

The two proposed heuristics for γ and μ are — in the majority of cases — consistent with



(a) Based on CLR weights. (b) Based on GENIE3 weights.

Figure 4.21 ~ Sensitivity analysis of μ and γ on the dataset 2 of DREAM4 ~ *Assessment of the parameter effects on AUPRs obtained using BRANE Cut on (a) CLR and (b) GENIE3 weights. For each γ, results obtained with BRANE Cut are given in terms of average AUPR and standard deviation over μ. BC*-CLR refers to the AUPR results obtained with BRANE Cut parametrized by the data-driven heuristic. The AUPR obtained with CT as also recalled.*

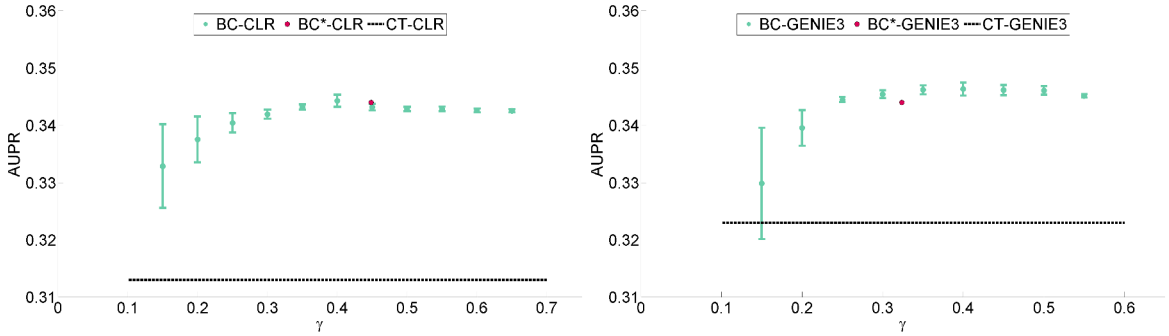


(a) Based on CLR weights. (b) Based on GENIE3 weights.

Figure 4.22 ~ Sensitivity analysis of μ and γ on the dataset 3 of DREAM4 ~ *Assessment of the parameter effects on AUPRs obtained using BRANE Cut on (a) CLR and (b) GENIE3 weights. For each γ, results obtained with BRANE Cut are given in terms of average AUPR and standard deviation over μ. BC*-CLR refers to the AUPR results obtained with BRANE Cut parametrized by the data-driven heuristic. The AUPR obtained with CT as also recalled.*

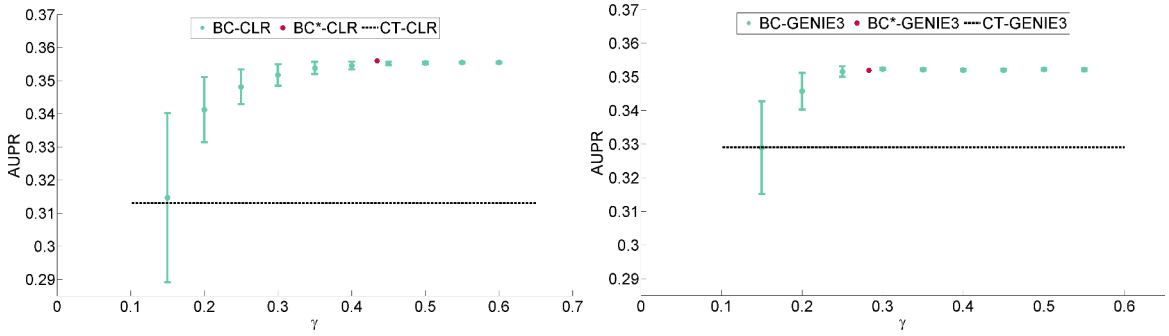
the order of magnitude parameters yielding maximal results. This data-driven parameter setting yields good compromise on tested datasets and offers a suitable start-point for parameter adjustment to refine results.

What is the computational complexity of BRANE Cut? We used the C++ code implementing a max-flow algorithm from Boykov and Kolmogorov (2004). Using this algorithm, the computational complexity of *BRANE Cut* is, in the worst-case, $O(mn^2|C|)$, where m (respectively n) is



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 4.23 ~ SENSITIVITY ANALYSIS OF μ AND γ ON THE DATASET 4 OF DREAM4 ~ *Assessment of the parameter effects on AUPRs obtained using BRANE Cut on (a) CLR and (b) GENIE3 weights. For each γ, results obtained with BRANE Cut are given in terms of average AUPR and standard deviation over μ. BC*-CLR refers to the AUPR results obtained with BRANE Cut parametrized by the data-driven heuristic. The AUPR obtained with CT as also recalled.*



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 4.24 ~ SENSITIVITY ANALYSIS OF μ AND γ ON THE DATASET 5 OF DREAM4 ~ *Assessment of the parameter effects on AUPRs obtained using BRANE Cut on (a) CLR and (b) GENIE3 weights. For each γ, results obtained with BRANE Cut are given in terms of average AUPR and standard deviation over μ. BC*-CLR refers to the AUPR results obtained with BRANE Cut parametrized by the data-driven heuristic. The AUPR obtained with CT as also recalled.*

the number of edges (respectively the number of nodes) in the flow network $\mathcal{G}_f$, and $|C|$ the cost of the minimal cut. Specifically, in our case — without the dimension reduction trick — the number of nodes n in $\mathcal{G}_f$ is equal to the sum of the number of edges E in the initial graph $\mathcal{G}$, the number of gene nodes G plus two additional nodes (the source and the sink). The order of magnitude for the number edges m in $\mathcal{G}_f$ is $G^2 + q$, where q is the number of edges coding for the co-regulation *a priori*. Note that, as mentioned in Boykov and Kolmogorov (2004),

this complexity is not the best achievable by a max flow algorithm. Meanwhile, their experiments showed better performance for several typical computer vision problems. Not being in a computer vision setting, we could benefit from faster max flow algorithms. However, since the time spent on max flow computation to infer the large graph of *Escherichia coli* is small (only several seconds), the benefit would not be noticeable. Given pre-computed weights, our algorithm requires 30 additional seconds to infer the E. coli network, without using the simplification described in the Section 4.1.2. By computing the explicit solution to our problem on a subset of edges, we improve *BRANE Cut* computation times by a factor of 10. Given CLR weights computed in 41 minutes on a Intel Core i7, 2.70 GHz laptop, our algorithm thus only requires three additional seconds. We note that the weight computation duration of GENIE3 is sensibly longer (5 h), using the list of transcription factors. If one wished to build an *E. coli* network that would also contain $\overline{\text{TF}}$-$\overline{\text{TF}}$ interactions using GENIE3, it would take 20 minutes per gene, for a total of two months with a basic rule of three.

Served by all the above benchmark validations and sensitivity analyses, we confidently can turn to the inference of *Trichoderma reesei*.

4.2 *BRANE Cut*: application on *Trichoderma reesei*

In this section, we briefly recall the essential knowledge we dispose regarding cellulase production mechanism by the fungus *Trichoderma reesei*. We then provide some preliminary results obtained by performing standard bio-informatics analyses. Their validation allows us to go further by the use of *BRANE Cut*.

4.2.1 Actual knowledge on *T. reesei* cellulase production system

As introduced in Section 1.1, the fungus *Trichoderma reesei* is a well-adapted micro-organism to produce cellulases — enzymes responsible for the degradation of the cellulose into glucose molecules. Its use in second generation biofuel process is thus natural. Understanding the functioning of such a fungus in the cellulase production context is a longstanding research field. Indeed, from several decades, several lineages of hyper- and hypo-producer strains have been generated using random mutagenesis (see Figure 4.25). Analyzing -omics data from this variety of strains can help us to better understand the regulatory mechanism of the cellulase production. Before focusing on genetic regulatory aspects, it is important to take an inventory of existing type of cellulases produced by *Trichoderma reesei*. For this purpose, Table 4.7 lists the main cellulolitic enzymes produced by the fungus and we refer to Foreman *et al.* (2003) for a more complete review of them.

We now give some words on the already identifyied regulatory mechanism for the cellulase production when *T. reesei* is induced by lactose and provide a non exhaustive literature for a more complete overview of the regulatory process. The transcription factor XYR1 (*xyr1*) has been identified as a pivotal inducer of cellulolitic enzymes production. Their production vanishes with its suppression (Stricker *et al.*, 2006, 2008; Mach-Aigner *et al.*, 2008). The transcription factor CRE1, responsible for the catabolite repression, is one of the most influent repressor of

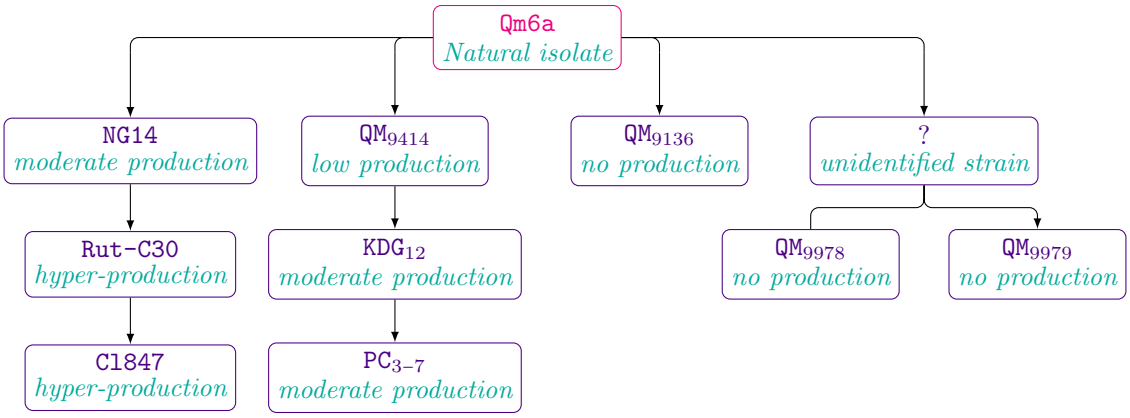


Figure 4.25 ~ LINEAGE OF *Trichoderma reesei* STRAINS ~
All strains are generated by random mutagenesis, essentially using NTG (N-methyl-N'-nitro-N-nitrosoguanine). Note that this genealogy is incomplete and non studied strains are not mentioned, notably one strain between `Qm6a` *and* `NG14`*, five strains between* `Rut-C30` *and* `C1847`*, one strain between* `Qm6a` *and* $\texttt{QM}_{9414}$*, and four strains between* $\texttt{QM}_{9414}$ *and* $\texttt{KDG}_{12}$*.*

the cellulase production. Indeed *cre1*-deleted strains reveal higher production levels (Nakari-Setälä *et al.*, 2009), as it is the case for the `Rut-C30` strain. Some studies have also reported a link between XYR1 and the catabolite repression (Strauss *et al.*, 1995; Seidl *et al.*, 2008; Portnoy *et al.*, 2011). Others TFs, such as ACE1, ACE2, ACE3, BGLR, or pMH29 have also been identified to be involved in the cellulase production process (Saloheimo, 2000; Aro *et al.*, 2001; Portnoy, 2011; Denton and Kelly, 2011; Seiboth *et al.*, 2012; Häkkinen *et al.*, 2014). Nevertheless, the precise role of such TFs remain — for the moment — enigmatic. In addition, interesting results are drawn from our previous study (Poggi-Parodi *et al.*, 2014) consisting in a transcriptomic comparison of strain `NG14` and `Rut-C30` during the cellulase induction process. Indeed, while a large number of mutations are found in hyper-producer strains, our study reveals that only a low number of transcription factors involves in the cellulase production is mutate suggesting an essentially intact induction system. Moreover, in the work by Jourdier *et al.* (2013), authors observed differential enzyme activities between β-glucosidases and cellulases according to the proportion of lactose inducer in a mixture of sugars as carbon source.

From this sparse knowledge, we proposed an experimental design to generate RNA-seq data allowing us to confirm, at the transcriptomic level, phenotypes observed on the enzyme activities and to refine assumptions on the regulatory pathway of the cellulase production.

4.2.2 Dataset and preludes

We now present results obtained *via* standard bioinformatics analyses on the RNA-seq data of *Trichoderma reesei* `Rut-C30` strain (Montenecourt and Eveleigh, 1977). Note that the detailed experimental protocol is described in Section 3.2.1. We recall that data are composed of read counts for 9129 genes in 36 experimental conditions, including various culture media — mixture

Function	ID	gene	protein	GH family
Exo-glucanase	**123989**	***cbh1/cel7a***	**CBH1/CEL7A**	GH7
	72567	***cbh2/cel6***	**CBH2/CLE6**	GH6
Endo-β-1,4-glucanase	**122081**	***egl1/cel7b***	**EG1/CEL7B**	GH7
	120312	***egl2/cel5a***	**EG2/CEL5A**	GH5
	123232	*egl3/cel12a*	EG3/CEL12A	GH12
	73643	*egl4/cel61a*	EG4/CEL61A	GH61
	49976	*egl5/cel45a*	EG5/CEL45A	GH45
	82616	*egl8/cel5b*	EG8/CEL5B	GH5
	49081	*cel74a*	CEL74A	GH74
	120961	*cel61b*	CEL61B	GH61
β-glucosidase	76672	*bgl1/cel3a*	BGL1/CEL3A	GH3
	120749	*bgl2/cel1a*	BGL2/CEL1A	GH1
	121735	*cel3b*	CEL3B	GH3
	82227	*cel3c*	CEL3C	GH3
	46816	*cel3d*	CEL3D	GH3
	76227	*cel3e*	CEL3E	GH3
	22197	*cel1b*	CEL1B	GH1

Table 4.7 ~ List of main cellulolitic enzymes of *T. reesei* ~
Glycosyl Hydrolase (GH) are classified in family, according to their amino acid sequence similarity determining a type of structure. The enzymes highlighted in bold are the four most abundant components among cellulases. Under inducing conditions, they may represent 50 % *of the produced proteins. We note that one specific function does not always involve one kind of structure, as revealed by the diversity of GH.*

of glucose and lactose in various proportions — and biological replicates. In this study, standard bioinformatics analyses (normalization, differential expression analysis (DE) and gene clustering) are required in order to validate the generated data by recovered known information from the literature. Once data are validated, it can thus be possible to go further by inferring the GRN with *BRANE Cut*.

~ Normalization, differential expression analysis and gene selection ~ The DESeq normalization is firstly carried out in order to compare the gene expression levels across the experimental conditions. A differential analysis were then performed to identify if the observed difference in read counts is significant. Both normalization and differential experession (DE) analysis was performed using the Bioconductor R package *DESeq* of Anders and Huber (2010) and described in Section 2.3. In addition, an adjustment for multiple-testing with the procedure of Benjamini and Hochberg (Benjamini and Hochberg, 1995) was also employed for the differential analysis. Specifically, to refine the knowledge of the lactose effect on the cellulase production, the gene expressions on various lactose concentration (G_{90}-L_{10}, G_{75}-L_{25}, L_{100}) at 24 h and 48 h are differentially evaluated regarding gene expression obtained on pure sugar e.g. glucose (G_{100})

or lactose (L_{100}) at 24 h and 48 h. The used methodology leads to ten pairwise comparisons, sketched on the circuit design displayed in Figure 4.26.

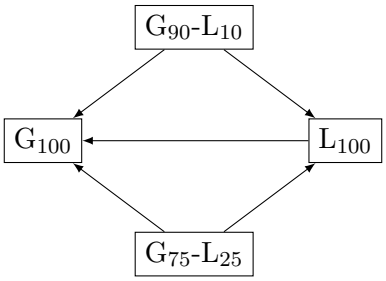


Figure 4.26 ~ CIRCUIT DESIGN FOR THE SEARCH OF DIFFERENTIALLY EXPRESSED GENES ~ *This design allows us to evaluate differential gene expressions across five comparisons. It is applied on the gene expression obtained at* 24 h *and* 48 h*, leading to ten comparisons.*

Based on this DE analysis, we assumed that a gene is said differentially expressed when the adjusted p-value was lower than 0.001 and the absolute value of the logarithm of FC was higher than 2. Here, FC refers to the fold-change of the read counts for the tested condition (G_{90}-L_{10}, G_{75}-L_{25}, or L_{100}) against the read counts for the reference condition (G_{100} or L_{100}). Using the chosen criteria, 650 genes are identified as differentially expressed in at least one of the ten studied comparisons. Figure 4.27 recaps the number of over- and underexpressed genes on various mixed carbon source media at 24 h and 48 h.

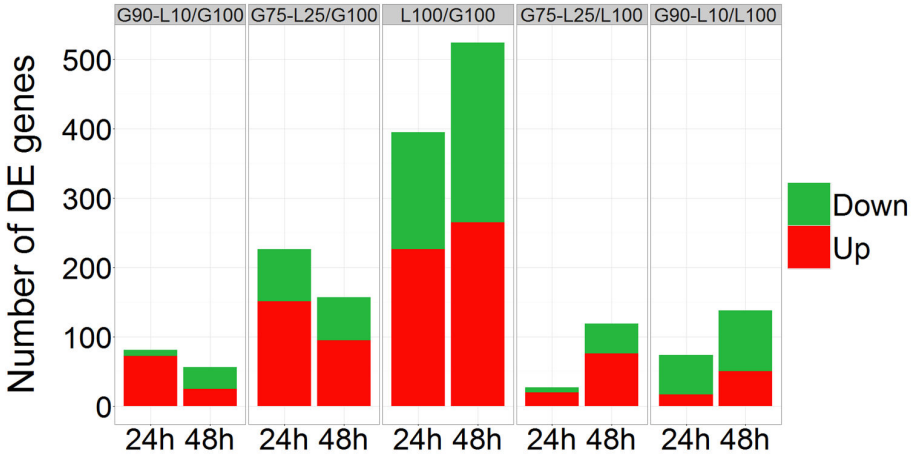


Figure 4.27 ~ DE GENES OF RUT-C30 ON VARIOUS MIXING OF CARBON SOURCES ~ *Number of over- (Up, in red) and under-(Down, in green) expressed genes on various mixing carbon source media at* 24 h *and* 48 h.

~ *Clustering analysis of differentially expressed genes* ~ These 650 genes only are thus used for a gene classification procedure, were genes are grouped according to similar profiles. In our study, we choose as gene profile the logarithm of the fold-change for the ten comparison. Fold-change are obtained by averaging read counts across the biological replicates in the tested and reference condition.

The following approach was completely performed using the MultiExperiment Viewer (MeV) software (Howe *et al.*, 2010). Firstly, a hierarchical clustering allows us to estimate the optimal number of clusters K containing in the data. By choosing a Euclidean distance metric and the average linkage method, results incited us to define K equal to 5. Then, the K-means algorithm is preferred in order to obtain a final gene classification. As this method is sensitive to initialization, we performed ten independent runs of K-means with random initialization, where for each run the Euclidean distance is used. Then, results are aggregated in order to be close to five consensus clusters. The aggregation is constrained by an occurrence threshold, fixed to 80 %. As a result, the 650 genes are completely classified into five clusters and no unassigned cluster was found. The five clusters, respectively denoted by C_1, C_2, C_3, C_4 and C_5, are composed of 254, 201, 78, 53 and 64 genes. For each cluster, the median gene expression profile is computed and results are displayed in Figure 4.28.

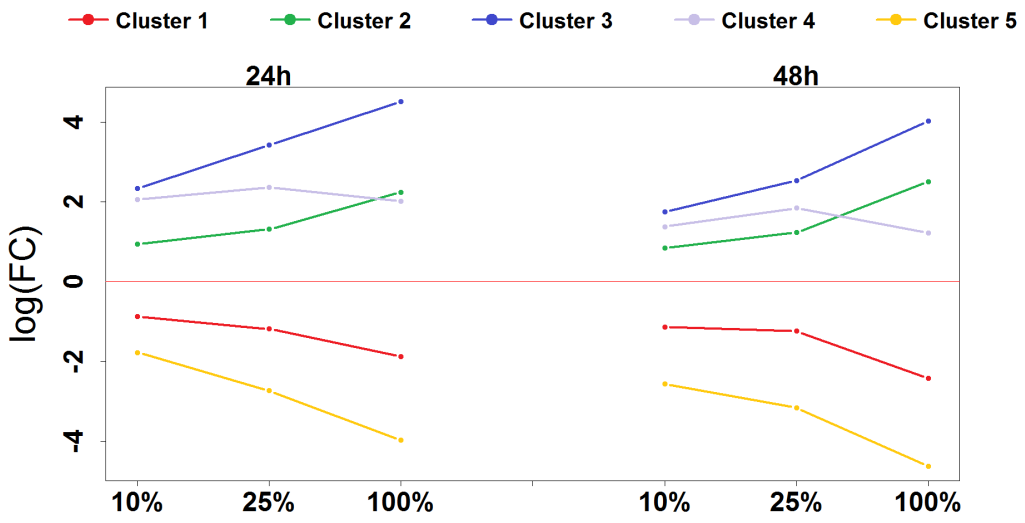


Figure 4.28 ~ MEDIAN PROFILES OF THE FIVE CLUSTERS OBTAINED FROM 650 DE GENES ~ *Median profile trends for differential expression levels (*log(FC)*) at* 10 %*,* 25 % *and* 100 % *of lactose with respect to pure glucose*

Classification results allow us to distinguish five distinct gene behaviors when the fungus feeds on lactose compared to its growth on glucose. Up to a scale factor, they can be described by three macroscopic trends. The first trend encompasses genes underexpressed on lactose, in a monotonic manner — more lactose implies less expression — at 24 h and 48 h (clusters C_1 and

C_5). On the contrary, the second one refers to genes overexpressed on lactose in a monotonic manner — more lactose implies more expression — at 24 h and 48 h (cluster C_2 and C_3). The last trend concerns genes overexpressed on lactose, but where the amount of lactose affects the gene expression in a quasi-stationary manner (cluster C_4). The functional enrichment analysis in each cluster reveals that underexpressed genes on lactose (C_1 and C_5) are mainly related to development and signaling pathway in addition to proteolysis and cell surface. Enriched genes showing an overexpression on lactose (C_2, C_3 and C_4) are — as expected — related to carbohydrate metabolism in addition to MFS[1] and carbohydrate transport. These preliminary results are coherent with the literature and allows us to go further with *BRANE Cut*.

We now present network inference results obtained with *BRANE Cut* using the *Trichoderma reesei* data, restricted to DE genes as for the clustering task.

4.2.3 New insights on cellulase production

For the network inference part, we choose a slightly modified version of this previous expression matrix, while keeping the same initial set of differentially expressed genes. Indeed, we preferentially deal with all biological replicates for the tested conditions while the reference conditions are pooled. In other words, the log fold-change is computed between the read count coming from a biological replicate of the test condition and the averaged read counts of the reference condition. Hence, for a given comparison, we obtained as many log fold-changes as biological replicates. In order to restrict the variability caused by this approach, we removed genes for which a biological replicate has a null read count. As a result, the final matrix contains 593 genes, where for each gene the expression profile contains 32 components. Although we incorporate variability, this procedure allows us to deal with expression profiles having a sufficient number of components to obtain a more reliable inferred network. We compute the complete weighted adjacency matrix thanks to CLR (Faith *et al.*, 2007). After normalizing these CLR weights between 0 and 1, we then use our *BRANE Cut* approach to obtain a GRN. The latter GRN was obtained using λ_{TF} and $\lambda_{\overline{\mathrm{TF}}}$ equal to 0.2 and 0.054, respectively — factor β is close to 3.7. Parameters controlling the co-regulation prior was set to 0.2 and 2 for γ and μ, respectively. The parameter γ given by the heuristic equals 0.36 and is thus close to the chosen one. Note that a factor ten is observed between the chosen β and μ parameters and those computed using the heuristics. Heuristics was validated on datasets where the proportion of TFs reaches, in average, 30 % of the total number of genes, In our *Trichoderma reesei* dataset, this proportion drop to only 3 %. Proposed heuristics have thus to be adapted for especially low proportion of TFs.

The resulting network contains 161 genes and 205 edges. In order to take advantage of classification results, we colored node according to the cluster it belongs. Doing this, we observe that modules (or sub-networks) in the whole network are coherent with clustering results, yielding a first validation of the inferred network. Network analysis is then carried out at two levels: validation of known or expected relationships and prediction. Despite the relatively poor knowledge

[1] MFS (major facilitator superfamily) is a superfamily of membrane transport proteins.

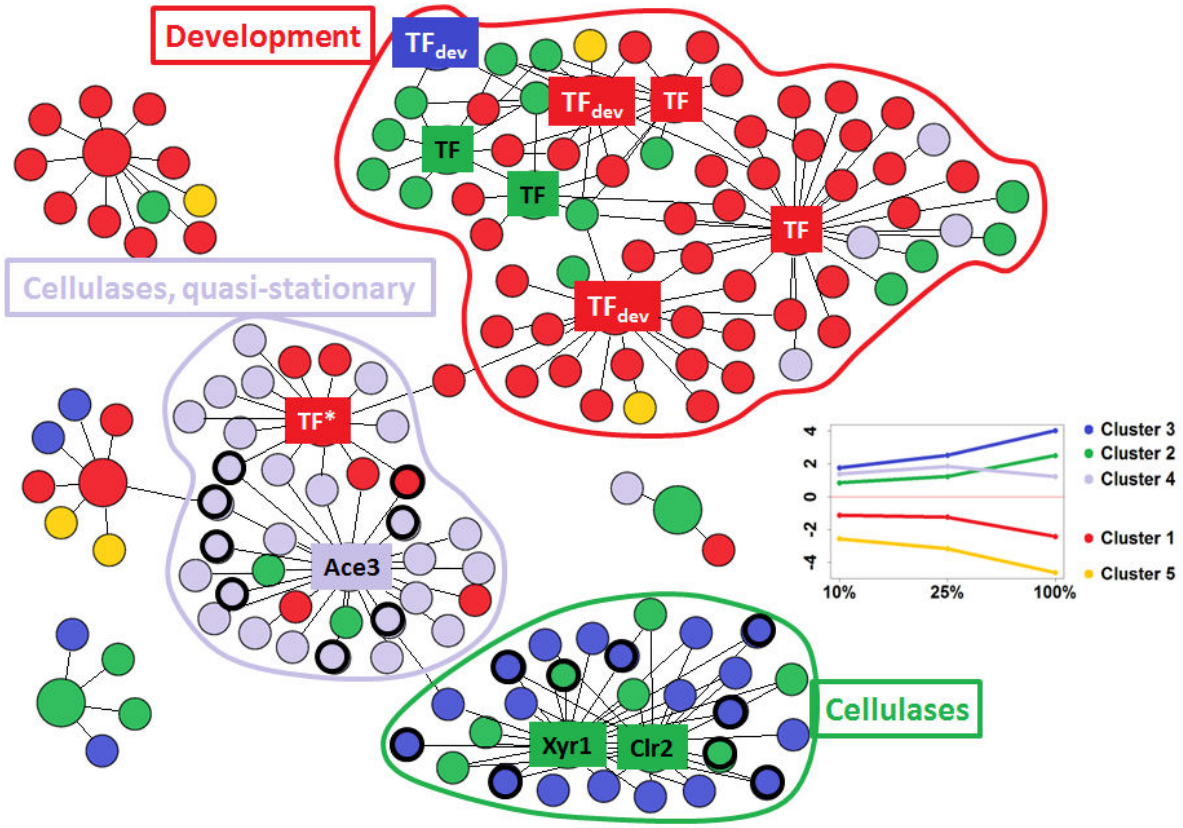


Figure 4.29 ~ *Trichoderma reesei* INFERRED NETWORK ~
Network built on 593 differentially expressed genes. It contains 161 genes and 205 edges. Node coloring corresponds to cluster labels: red (C_1), green (C_2), blue (C_3), purple (C_4) and yellow (C_5). Bigger nodes correspond to genes coding for a transcription factor while smaller nodes correspond to genes not identified to code for a transcription factor.

on regulatory mechanism regarding cellulase production and the fact that about 27 % of genes present in the network have no identified function, some clues allow us to validate the network and give confidence for further biological assumptions. First of all, the 161 selected genes — including 15 TFs — only cover a relatively small number of biological processes. Specifically, a significant proportion is reliably supposed to be involved in the cellulase production and development. On the one hand, we recover the cellulase-related TFs. In addition, 17 cellulolitic enzymes (among the 35 identified by Foreman *et al.* (2003)) are recovered in the network. On the other hand, we found four development-related TFs in addition to five other genes. We also observe numerous genes related to transport and secretory systems. In details, 12 transport protein are recovered while 14 genes coding for secreted proteins are present in the network.

These genes are mainly arranged in coherent modules allowing us to distinguish three interesting sub-networks as highlighted in Figure 4.29. The first sub-network (circled in green)

encompasses the main cellulolytic enzymes and their associated transcription factor XYR1 (ID 122208) in addition to secreted proteins and transporters. All genes involved in this sub-network belong to clusters C_2 and C_3 and thus share a monotonic over-expression profile. The second sub-network, circled in purple, mainly contains genes coding for proteins involved in the carbohydrate metabolism — and notably the β-glucosidases — and are linked to the transcription factor ACE3. They belong to cluster C_4, characterized by a quasi-stationary over-expression profile. Some of them are also linked to the TF with pMH29, which interestingly has an inverse profile of *ace3*. Finally, the third and last sub-network, circled in red, embraces genes related to development process and belonging to cluster C_1. We also found, in this sub-network, genes pertaining to carbohydrate metabolism. These relationships suggest that, in presence of lactose, a link — albeit indirect — exists between cellulase production induction and development repression. Based on the observed sub-networks, Table 4.8 summarizes some elements of the literature allowing us to validate the inferred whole network by *BRANE Cut*.

Gene ID	Name	Up/Down	Link to CP	Specie	Reference
122208	*xyr1*	up	direct	*T. reesei*	Stricker *et al.* (2006)
26163	*clr2*	up	direct	*N. crassa*	Coradetti *et al.* (2012)
77513	*ace3*	up	direct	*T. reesei*	Häkkinen *et al.* (2014)
122523	*pmh29*	down	direct	*T. reesei*	Häkkinen *et al.* (2014)
123713	*medA*	down	indirect	*P. decumbuns*	Qin *et al.* (2013)
76590	*pro1*	down	direct	*P. oxalicum*	Zhao *et al.* (2016b)
4430	*wetA*	down	indirect	*P. decumbuns*	Qin *et al.* (2013)

Table 4.8 ~ *BRANE Cut* NETWORK VALIDATION FROM LITTERATURE **~**

In light of the presented element, we consider the network inferred by *BRANE Cut* as reliable and finer analysis can be performed in order to extract some new insight on the cellulase production mechanisms. Indeed, while the sub-network concerning cellulases is expected, the presence of the gene *clr2* at the same level of gene *xyr1* is a probable insight to be validated. In addition, one of the main assumptions issued from this network is the potential link between cellulase production and development process. While some clues in favor of this link are found in other fungus species, its manifestation in *Trichoderma reesei* is poorly studied. In order to validate such suggested links between development and cellulase production, it can be judicious to proceed to genetic engineering on well-chosen development-related TFs such as gene ID 76590 or 102499. Regarding the differential expression of gene ID 102499 and 76590 we chose to prepare two kinds of `Rut-C30` mutants: one with a deletion of gene ID 102499, the other overexpressing (Prelich, 2012) gene ID 76590. Preliminary results in well on plate for the two above mutants suggest an influence of these two genes on the cellulase production. Additional experiments in flask are in progress to confirm their influence. Moreover, we confirm, at the transcriptomic scale, the differential enzyme activities between β-glucosidases and cellulases with respect to the lacostose inducer concentration. Combining phenotypic and transcriptomic results from the inferred network, we may assume that distinct regulatory pathways for the β-glucosidases and the cellulases exist.

4.3 Conclusions on *BRANE Cut*

BRANE Cut is our first edge selection strategy for GRN refinement. Its design favors the selection of strongly weighted edges in addition to two biological priors enforcing network modularity and gene co-regulation. The formulation is an instance of a minimum cut energy function and is solved using a maximal flow algorithm. The latter is applied on a transportation network which can be viewed as the dual network of the initial complete graph. Numerical improvements over state-of-the-art are recovered in both synthetic and real datasets from the DREAM4 and DREAM5 challenges. Biological relevance of inferred GRNs was validated on both *Escherichia coli* and *Trichoderma reesei* networks.

5

Edge selection refinement using gene connectivity a priori *(BRANE Relax)*

"The power of mathematics is often to change one thing into another, to change geometry into language"

Marcus du Sautoy

This chapter is dedicated to the presentation of BRANE Relax published in (Pirayre et al., 2015b). This approach was designed to perform edge selection on a complete weighted network for GRN inference. Integrating biological a priori *regarding the connectivity of particular genes, the constrained optimization problem we formulat can be relaxed into a convex one for which proximal algorithms can be used. Taking into account the high dimensionality of the problem, recent tricks for algorithm acceleration such as pre-conditioning and a block coordinate scheme are used. Comparative results on standard simulated datasets from the DREAM4 and DREAM5 challenges demonstrate substantial improvements over conventional approaches.*

Contents

As introduced in Section 3.3.1, our prior-based edge selection strategy aims at defining an objective function to be optimized — depending on binary variables $x_{i,j}$ reflecting the presence/absence of the edge $e_{i,j}$ in the GRN to be inferred. In the following, we thus detail how to choose appropriate cost function, inspired by the optimization formulation of the classical thresholding in (3.24), to encode some biological *a priori* exposed in Section 5.1.1. We recall here the optimization formulation of the classical thresholding that we adapt to lead to *BRANE Relax*:

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{minimize}} \quad \sum_{\substack{(i,j)\in\mathbb{V}^2 \\ j>i}} \omega_{i,j}\,(1-x_{i,j}) \quad + \quad \lambda\, x_{i,j}, \tag{5.1}$$

where the notation is the same as the previous chapter.

5.1 *BRANE Relax* problem formulation

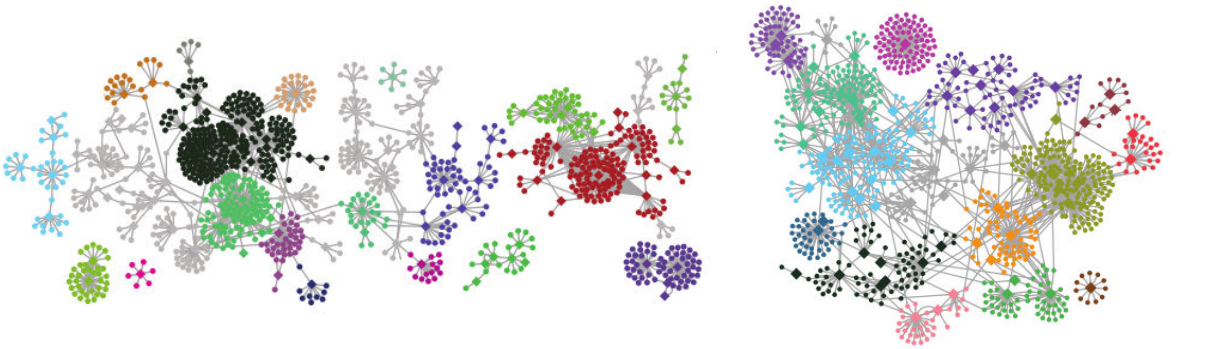
5.1.1 Gene connectivity *a priori*

We first recall some biological background justifying our *BRANE* methodology. Gene regulation is a complex mechanism involving lots of entities at various scales of the cell behavior: DNA, RNA, proteins, chromatin condensation, etc. Nevertheless, the main actors in gene regulation are transcription factors (TFs) i.e. proteins regulating gene expression. The availability of such a set of TFs often results from the combination of dedicated experiments to identify them and a knowledge from the literature or stored in (public) databases. Moreover, some yet unvalidated TFs are also predicted as such thanks to the presence of specific DNA-binding patterns in their sequence. It is thus common to hold such a list of TFs, and, being the main actors of the gene regulation, biological *a priori* can be established from them. We thus focus our work by considering two kind of genes, metonymically referred to as TFs and $\overline{\text{TF}}$s (the latter denotes genes not identified to code for a transcription factor). Note that knowing which gene is a TF does not provide information on which genes it acts.

Among the TFs, various levels of action appear. Some TFs — involved in general mechanism such as transcription, translation, etc. — can regulate the expression of hundred of genes. Conversely, others TFs are extremely specific and regulate a very small number of genes. Between these two extremes, a variety of mechanisms exist. It can thus become obvious that, having at our disposal only information about which genes are TFs, the formulation of an *a priori* on the TFs connectivity can turn out to be inappropriate. However, a prior on the connectivity of the $\overline{\text{TF}}$s can be considered. Indeed, we can assume that, without too much misuse, a $\overline{\text{TF}}$s is generally regulated by a small number of TFs. This biological *a priori* can thus be used to improve the GRN inference process.

Which effect of the proposed connectivity a priori on the GRN? As hereinabove explained, for a given $\overline{\text{TF}}$, we want to control the number of TFs acting on it, while no constraint is formulated on the number of $\overline{\text{TF}}$s which a TFs should regulate. In a graph structure $\mathcal{G}$, where nodes represent genes, this *a priori* is equivalent to constraining the degree of $\overline{\text{TF}}$ nodes to be close to a given small number d, while the degree of TF nodes is not particularly controlled. Such an *a priori*

models modular networks — a structure typically observed on GRN, as illustrated in Figure 5.1.



(a) E. coli *community network.* *(b)* S. cerevisae *community network.*

Figure 5.1 ~ GRNs WITH MODULAR STRUCTURE ~
Gene regulatory network of (a) Escherichia coli *and (b)* Saccharomyces cerevisae *obtained by combining predictions of the DREAM5 challengers. Illustrations adapted from* Marbach et al. *(2012) - p. 6 (some text mentions have been removed for clarity).*

In order to introduce our *a priori*, we recall that $\mathbb{V}$ denotes the set of node (gene) indices and $\mathbb{T}$ the set of TF nodes indices only. Now, assuming $x_{i,j}$ is a binary label reflecting the presence/absence of edge $e_{i,j}$, the degree of a $\overline{\text{TF}}$ node i, for each $i \in \mathbb{V}\backslash\mathbb{T}$, is evaluated by summing the labels $x_{i,j}$, for all $j \in \mathbb{V}$. Hence, the constraint on the degree of the $\overline{\text{TF}}$ nodes can thus be mathematically encoded through a regularization term defined as follows:

$$\psi(x_{i,j}) = \sum_{i \in \mathbb{V}\backslash\mathbb{T}} \phi\left(\sum_{j \in \mathbb{V}} x_{i,j} - d\right), \tag{5.2}$$

where the function ϕ is a convex function, with β-Lipschitz continuous gradient, quantifying, for each $\overline{\text{TF}}$ node, the difference between its degree and a fixed small number d.

In addition, as opted to in *BRANE Cut*, modular structures can also be re-enforced by defining the regularization parameter λ associated to the second term in (3.24) according to the nature of nodes i and j. We refer to our previous Section 4.1.1 in which a detailed description of this prior is given. We recall here that this additional *a priori* is formulated through

$$\lambda_{i,j} = \begin{cases} 2\eta & \text{if } (v_i, v_j) \notin \mathcal{T}, \\ 2\lambda_{\text{TF}} & \text{if } (v_i, v_j) \in \mathcal{T}, \\ \lambda_{\text{TF}} + \lambda_{\overline{\text{TF}}} & \text{otherwise.} \end{cases} \tag{5.3}$$

Biological *a priori* now being introduced and modeled, we now describe the whole *BRANE Relax* formulation for network edge selection in a GRN context.

5.1.2 Initial formulation and relaxation

Combining the selection of strongly weighted edges — parametrized by the threshold $\lambda_{i,j}$ as in the classical thresholding — and our biological *a priori* on the connectivity of $\overline{\text{TF}}$s genes, our optimization problem is expressed as

$$\underset{\boldsymbol{x}\in\mathbb{S}}{\text{minimize}} \quad \sum_{\substack{(i,j)\in\mathbb{V}^2\\ j>i}} \frac{\omega_{i,j}}{2}\,(1-x_{i,j}) \;+\; \frac{\lambda_{i,j}}{2}\,x_{i,j} \;+\; \mu \sum_{i\in\mathbb{V}\setminus\mathbb{T}} \phi\left(\sum_{i\in\mathbb{V}} x_{i,j} - d\right), \tag{5.4}$$

where $\mu \in [0, +\infty[$ is a regularization constant controlling the impact of our connectivity prior on the edge selection, and

$$\mathbb{S} = \{(x_{i,j})_{(i,j)\in\mathbb{V}^2} \in \{0,1\}^E \mid \left(\forall (i,j) \in \mathbb{V}^2\right)\; x_{i,j} = x_{j,i}\}. \tag{5.5}$$

The latter constraint set serves to express both the Boolean constraint and the fact that the graph is undirected (symmetric weights $\omega_{i,j}$). In such a case, a symmetry property on $\lambda_{i,j}$ also has to be assumed:

$$\forall (i,j) \in \mathbb{V}^2, \qquad \lambda_{i,j} = \lambda_{j,i}. \tag{5.6}$$

Nevertheless, the cost function of Problem (5.4) is not necessarily sub-modular. It is thus not amenable to optimization via efficient combinatorial optimization methods such as Graph Cuts based methods. To overcome this difficulty, we relax the integrality constraint on $\boldsymbol{x}$, by replacing $\mathbb{S}$ by its convex hull:

$$\hat{\mathbb{S}} = \{(x_{i,j})_{(i,j)\in\mathbb{V}^2} \in [0,1]^E \mid \left(\forall (i,j) \in \mathbb{V}^2\right)\; x_{i,j} = x_{j,i}\}. \tag{5.7}$$

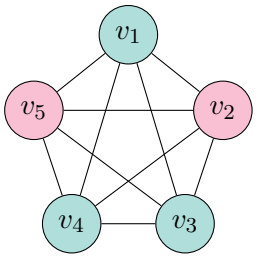
The relaxed optimization problem then becomes solvable in an efficient manner by using convex optimization methods which details are now provided.

5.2 *BRANE Relax*: optimization *via* a proximal framework

The relaxed optimization problem can be re-expressed more concisely by re-indexing the variables on the edges with a single index $l \in \{1, \ldots, E\}$, where $E = G(G-1)/2$ as we explicitly take into account the symmetry constraint. In such a case, edge labels $(x_1, x_2, \ldots, x_E)$ are equivalent to $(x_1,2,\, x_1,3,\, \ldots,\, x_{G-1,G})$. Using the vectorial formulation of the edge labels, the degree of a $\overline{\text{TF}}$ node v_i can be computed thanks to a binary linear operator $\boldsymbol{\Omega} \in \{0,1\}^{P\times E}$, where P is the number of $\overline{\text{TF}}$ nodes i.e. the cardinality of $\mathcal{V}\setminus\mathcal{T}$. This operator — reflecting the connection in the complete graph — is defined, for all $i \in \{1, \ldots, P\}$ and $j \in \{1, \ldots, E\}$, as follows

$$\Omega_{i,j} = \begin{cases} 1 & \text{if } j \text{ is the index of an edge linking the } \overline{\text{TF}} \text{ node } v_i \text{ in the complete graph,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

Let us give, in Figure 5.2, an explicit construction of the matrix $\boldsymbol{\Omega}$ from a toy example.



(a) Complete graph.

$$\mathbf{\Omega} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \qquad \mathbf{\Omega}\boldsymbol{x} = \begin{pmatrix} x_1 + x_2 + x_3 + x_4 \\ x_2 + x_5 + x_6 + x_9 \\ x_3 + x_6 + x_8 + x_{10} \end{pmatrix}$$

(b) Construction of $\mathbf{\Omega}$. (c) Degree computation.

Figure 5.2 ~ CONSTRUCTION OF THE DEGREE MATRIX $\mathbf{\Omega}$ ~
On the complete graph in (a), pink and green nodes refers to TF *and* $\overline{\text{TF}}$ *nodes, respectively. We thus have* $P = 3$ *and* $E = 10$. *Based on this graph, the operator* $\mathbf{\Omega}$ *is constructed thanks to rules given in (5.8). Degree of* $\overline{\text{TF}}$ *nodes at the current values of* $\boldsymbol{x} = [x_1, \dots, x_{10}]^\top$ *is obtained thanks to the matrix product* $\mathbf{\Omega}\boldsymbol{x}$. *We recall here that the vectorial indexing* x_1, x_2, *etc. encodes the edge labels* $x_{1,2}$, $x_{1,3}$, *etc.*

The relaxed optimization problem becomes

$$\underset{\boldsymbol{x}\in[0,1]^E}{\text{minimize}} \quad \sum_{l=1}^{E} \big(\omega_l\,(1 - x_l) + \lambda_l\,x_l\big) \;+\; \mu \sum_{i=1}^{P} \phi\left(\sum_{k=1}^{E} \Omega_{i,k} x_k - d\right), \tag{5.9}$$

or in a equivalent vector form:

$$\underset{\boldsymbol{x}\in[0,1]^E}{\text{minimize}} \quad \boldsymbol{\omega}^\top (\mathbf{1}_E - \boldsymbol{x}) \;+\; \boldsymbol{\lambda}^\top \boldsymbol{x} \;+\; \mu\,\Phi(\mathbf{\Omega}\boldsymbol{x} - \mathbf{d}). \tag{5.10}$$

Hereabove, vectors $\boldsymbol{\omega}$, $\boldsymbol{\lambda}$ and $\boldsymbol{x}$ gather, for all $l \in \{1, \dots, E\}$, all variables ω_l, λ_l and x_l, respectively. In addition, $\mathbf{1}_E = [1, \dots, 1]^\top \in \mathbb{R}^E$ and $\mathbf{d} = d\mathbf{1}_P$, where $\mathbf{1}_P$ is defined analogously to $\mathbf{1}_E$. In the following, we assume Φ separable:

$$\Phi : \mathbb{R}^P \to \mathbb{R} : (y_i)_{1\le i\le P} \mapsto \sum_{i=1}^{P} \phi(y_i), \tag{5.11}$$

where $\phi{:}\,\mathbb{R} \to \mathbb{R}$ will be assumed convex and differentiable with a Lipschitzian gradient. As introduced in Section 3.3.4 through Equation (3.46), the constrained Problem (5.10) can be equivalently re-formulated into

$$\underset{\boldsymbol{x}\in\mathbb{R}^E}{\text{minimize}} \quad \boldsymbol{\omega}^\top (\mathbf{1}_E - \boldsymbol{x}) \;+\; \boldsymbol{\lambda}^\top \boldsymbol{x} \;+\; \mu\,\Phi(\mathbf{\Omega}\boldsymbol{x} - \mathbf{d}) \;+\; \iota_{[0,1]^E}(\boldsymbol{x}), \tag{5.12}$$

where $\iota_{[0,1]^E}(\boldsymbol{x})$ is the indicator function of the unit hypercube defined as:

$$\iota_{[0,1]^E}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in [0,1]^E, \\ +\infty & \text{otherwise.} \end{cases} \tag{5.13}$$

A proximal splitting strategy can be employed to re-express the optimization problem (5.12) as the minimization of a sum of two functions f_1 and f_2 such that f_1 belongs to $\Gamma_0(\mathbb{R}^E)$ — the set

of proper, lower semi-continuous and convex functions — and f_2 is a convex and differentiable function with a L-Lipschitz continuous gradient. Abiding by the previous rules, we define f_1 as the indicator function of the convex set $[0,1]^E$ i.e. $f_1(\boldsymbol{x}) = \iota_{[0,1]^E}(\boldsymbol{x})$ and

$$f_2(\boldsymbol{x}) = \boldsymbol{\omega}^\top (\mathbf{1}_E - \boldsymbol{x}) \; + \; \boldsymbol{\lambda}^\top \boldsymbol{x} \; + \; \mu\, \Phi(\boldsymbol{\Omega}\boldsymbol{x} - \mathbf{d}). \tag{5.14}$$

This scheme answers the requirements for the use of the Forward-Backward (FB) algorithm (details in Section 3.3.4) for which iterations are given by

$$\forall k \in \mathbb{N} \quad \boldsymbol{x}_{k+1} = \operatorname{prox}_{\gamma_k, f_1}\left(\boldsymbol{x}_k - \gamma_k \nabla f_2(\boldsymbol{x}_k)\right), \tag{5.15}$$

where for all $k \in \mathbb{N}$, the step-size γ_k belongs to $]0, 2(\mu L)^{-1}[$. However, in view of the dimension of the problem — potentially reaching hundreds of thousands of variables to be optimized — this first-order method can become pretty slow. We thus now present two tricks used to provide an accelerated version.

5.2.1 Preconditioning

The first strategy we used to accelerate the convergence rate of our FB algorithm relies on the Majorize-Minimize (MM) principle, for which details are provided in Section 3.3.5. We thus apply the MM principle to f_2 by building a quadratic majorant of this smooth function. For this purpose, we used the descent lemma introduced in Bauschke and Combettes (2011) with a variable metric. In our application, assuming that β is the Lipschitz constant of the function Φ, we have for every $(\boldsymbol{x}, \boldsymbol{x}') \in \mathbb{R}^E$

$$f_2(\boldsymbol{x}) \; \leq \; f_2(\boldsymbol{x}') \; + \; (\boldsymbol{x} - \boldsymbol{x}')^\top \nabla f_2(\boldsymbol{x}') \; + \; \frac{\mu\beta}{2} (\boldsymbol{x} - \boldsymbol{x}')^\top \boldsymbol{\Omega}^\top \boldsymbol{\Omega} \, (\boldsymbol{x} - \boldsymbol{x}'), \tag{5.16}$$

yielding a quadratic majorant function of f_2 at $\boldsymbol{x}'$ such that

$$Q(\boldsymbol{x}, \boldsymbol{x}') = f_2(\boldsymbol{x}') \; + \; (\boldsymbol{x} - \boldsymbol{x}')^\top \nabla f_2(\boldsymbol{x}') \; + \; \frac{\mu\beta}{2} (\boldsymbol{x} - \boldsymbol{x}')^\top \mathbf{A} \, (\boldsymbol{x} - \boldsymbol{x}'), \tag{5.17}$$

where $\mathbf{A}$ is a symmetric positive definite matrix majorizing $\boldsymbol{\Omega}^\top \boldsymbol{\Omega}$, i.e. such that $\mathbf{A} - \boldsymbol{\Omega}^\top \boldsymbol{\Omega}$ is semi-definite positive. Instead of directly minimizing $f_1 + f_2$, we design our optimization algorithm to minimize, at iteration k, the surrogate function $f_1 + \mathcal{F}(\cdot, \boldsymbol{x}_k)$. In such a case, based on (5.15), the Preconditioned Forward-Backward (P-FB) iteration is given by

$$\forall k \in \mathbb{N}, \quad \boldsymbol{x}_{k+1} = \operatorname{prox}_{\gamma_k^{-1}\mathbf{A}, f_1}(\boldsymbol{x}_k - \gamma_k \mathbf{A}^{-1} \nabla f_2(\boldsymbol{x}_k)), \tag{5.18}$$

where, for more flexibility, we have substituted a parameter $\gamma_k \in]0, +\infty[$ for the factor $(\mu\beta)^{-1}$. The proximity operator of function $\gamma_k f_1$ relative to the metric induced by $\mathbf{A}$ is given by

$$\forall \boldsymbol{x} \in \mathbb{R}^E, \quad \operatorname{prox}_{\gamma_k^{-1}\mathbf{A}, f_1}(\boldsymbol{x}) = \arg\min_{\boldsymbol{z} \in \mathbb{R}^E} \; \gamma_k f_1(\boldsymbol{z}) + \frac{1}{2} \|\boldsymbol{z} - \boldsymbol{x}\|_{\mathbf{A}}^2, \tag{5.19}$$

where $\|\cdot\|_{\mathbf{A}}$ is the weighted norm of $\mathbb{R}^E$ defined as

$$\forall \boldsymbol{z} \in \mathbb{R}^E, \quad \|\boldsymbol{z}\|_{\mathbf{A}} = \left(\boldsymbol{z}^\top \mathbf{A} \boldsymbol{z}\right)^{\frac{1}{2}}. \tag{5.20}$$

As above-mentioned, our aim is to define the matrix $\mathbf{A}$ as an approximation of $\mathbf{\Omega}^\top\mathbf{\Omega}$ — a scaled version of the Hessian of the function f_2 at $\boldsymbol{x}_n$. As it can be observed in (5.18), P-FB iteration requires the inverse of the matrix $\mathbf{A}$ which can appear cumbersome for large-size matrices. To circumvent this difficulty, a simple structure for the matrix $\mathbf{A}$ relying on a diagonal form can be employed. For this purpose, we used the construction rule proposed in Chouzenoux *et al.* (2014). The diagonal preconditioning matrix we obtained is thus:

$$\mathbf{A} = \mathrm{Diag}\left(\mathbf{R}^\top \mathbf{1}_P\right), \tag{5.21}$$

where $\mathbf{1}_P = [1, \ldots, 1] \in \mathbb{R}^P$ and $\mathbf{R} = \left(R_{i,k}\right)_{1\leq i\leq P,\, 1\leq k\leq E}$ with for every $i \in \{1, \ldots, P\}$ and $k \in \{1, \ldots, E\}$

$$R_{i,k} = \Omega_{i,k} \sum_{l=1}^{E} \Omega_{i,l}. \tag{5.22}$$

Due to the construction rules inherent to the definition of $\mathbf{\Omega}$ in (5.8), we can observe that, for all $i \in \{1, \ldots, P\}$, summing the E columns of the i-th row yields a constant number equals to $G-1$, where we recall that G is the number of genes. Hence, elements of $\mathbf{R}$ can be re-expressed, for every $i \in \{1, \ldots, P\}$ and $k \in \{1, \ldots, E\}$, as follows

$$R_{i,k} = \begin{cases} G-1 & \text{if } \Omega_{i,k} = 1, \\ 0 & \text{if } \Omega_{i,k} = 0. \end{cases} \tag{5.23}$$

Finally, the l-th diagonal element of $\mathbf{A}$, with $l \in \{1, \ldots, E\}$, can take only three values according to the nature of the edge indexed by l:

$$A_{l,l} = \begin{cases} 2(G-1) & \text{if } l \text{ is the index of an edge between two TFs,} \\ G-1 & \text{if } l \text{ is the index of an edge between a TF and a } \overline{\text{TF}}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.24}$$

As a result, the use of such preconditioning matrices allows us to increase the convergence speed of the algorithm, in terms of the number of iterations. In addition to a preconditioning of the FB algorithm, another strategy can be employed to accelerate the algorithm.

5.2.2 Block-coordinate descent strategy

As our objective function has been decomposed into a sum of a differentiable function f_2 and an additively separable function f_1, an improvement of the convergence speed can be expected by resorting to a block coordinate approach (Chouzenoux *et al.*, 2016). Indeed, from the separability hypothesis, an efficient alternating optimization scheme can be considered. At each iteration of the algorithm, it consists in updating a subset of variables only, while the others remain unchanged. While the previously detailed preconditioning strategy reduces the number of iterations, a block coordinate strategy allows to decrease the computational cost within an iteration. Combining both preconditioning and block coordinate approaches may drastically improve global convergence speed, especially for high-dimensional data.

For this purpose, assuming E variables to be optimized, we define $(\mathcal{P}_j)_{1\leq j\leq J}$ as a partition of $\{1,\dots,E\}$ into $J > 2$ subsets of cardinality Q, such that $E = JQ$. For each block index $j \in \{1,\dots,J\}$, $\mathcal{P}_j$ — the j-th element of the partition — corresponds to the set of indices defining a block of variables $\boldsymbol{x}_k^{(j)} \in \mathbb{R}^Q$ which may be activated at iteration k of the algorithm. The remaining $E - Q$ variables are unchanged. The j-th element $\mathcal{P}_j$ can be simply equal to $\mathcal{P}_j = \{Q(j-1)+1,\dots,jQ\}$.

We now focus on the block sweeping strategy — *which partition index j should be chosen at each iteration k?* — for which three main approaches exist. The *cyclic rule* is defined such that, for all iterations $k \in \mathbb{N}$ of the algorithm, $j_k - 1 = k \mod (J)$. The *quasi-cyclic rule* firstly introduced in Luo and Tseng (1992) generalizes the cyclic rule. The *quasi-cyclic rule* assumes that it exists a constant $K \geq J$ such that, for all iterations $k \in \mathbb{N}$, we have $\{1,\dots,J\} \subset \{j_k,\dots,j_{k+K-1}\}$. In such a case, blocks of variables can be updated in an arbitrary order if every block of variables is called in a finite number of iterations. Finally, in the *uniformly random rule*, the partition index j_k, at iteration $k \in \mathbb{N}$, is chosen such that j_k is a realization of a uniform random variable on $\{1,\dots,J\}$. In our work, the block sweeping strategy is chosen to follow a quasi-cyclic rule, thus guarantying our algorithm to converge to a (global) minimizer (Chouzenoux *et al.*, 2014).

Complemental to reducing the number of variables updated at each iteration, both the gradient computation and the preconditioning matrix benefit from a block coordinate strategy. Indeed, the gradient computation is performed with respect to the reduced-size vector $\boldsymbol{x}_k^{(j)} \in \mathbb{R}^Q$ only. This restriction implies the use of the sub-matrix $\boldsymbol{\Omega}_j$ of $\boldsymbol{\Omega}$ of dimension $P \times Q$ corresponding to the activated edges only. In the same vein, a more adapted preconditioning matrix $\mathbf{A}_j \in \mathbb{R}^{Q\times Q}$ can be employed. The reduced matrix corresponds to a diagonal majorizer of $\boldsymbol{\Omega}_j^\top \boldsymbol{\Omega}_j$ and is defined in a similar way as in (5.24).

Our *BRANE Relax* approach can thus be solved using a Block Coordinate Preconditioned Forward-Backward (BC-P-FB) algorithm, summarized in Algorithm 1.

Algorithm 1: *BRANE Relax*

Fix $\boldsymbol{x}_0 \in \mathbb{R}^E$;
for $k = 0,\ 1,\ \dots$ **do**
  Select the index $j_k \in \{1,\dots,J\}$ of a block of variables;
  $\boldsymbol{z}_k^{(j_k)} = \boldsymbol{x}_k^{(j_k)} - \gamma_k \mathbf{A}_{j_k}^{-1} \nabla_{j_k} f_2(\boldsymbol{x}_k)$;
  $\boldsymbol{x}_{k+1}^{(j_k)} = \mathrm{prox}_{\gamma_k^{-1}\mathbf{A}_{k_n}, f_1^{(j_k)}}(\boldsymbol{z}_k^{(j_k)})$;
  $\boldsymbol{x}_{k+1}^{(\bar{j_k})} = \boldsymbol{x}_k^{(\bar{j_k})}$, $\bar{j_k} = \{1,\dots,J\}\backslash\{j_k\}$.

For every $\boldsymbol{x} \in \mathbb{R}^E$ and $j \in \{1,\dots,J\}$, $\nabla_j f_2(\boldsymbol{x})$ is the partial gradient of f_2 with respect to $\boldsymbol{x}^{(j)}$ computed at $\boldsymbol{x}$. The above algorithm involves the computation of the proximity operator $\mathrm{prox}_{\gamma_k^{-1}\mathbf{A}_{\mathbf{j_k}}, f_1^{(j_k)}}$. It is reduced to the projection onto the convex set $[0,1]^Q$. In this context, the

proposed algorithm thus reduces to a block-coordinate variable metric variant of a projected gradient algorithm. In addition, the sequence of step-sizes $(\gamma_k)_{k\in\mathbb{N}}$ must be chosen such that

$$\inf_{k\in\mathbb{N}} \gamma_k > 0, \qquad \text{and} \qquad \sup_{k\in\mathbb{N}} \gamma_k < \frac{2}{\mu\beta}. \tag{5.25}$$

Note that our proposed algorithm returns the optimal edge labeling $\boldsymbol{x}^* \in [0,1]^E$, corresponding to the convex relaxation of our original problem. A last threshold at 0.5 is thus finally applied on the so-obtained minimizer to obtain the list of edges present in the inferred graph.

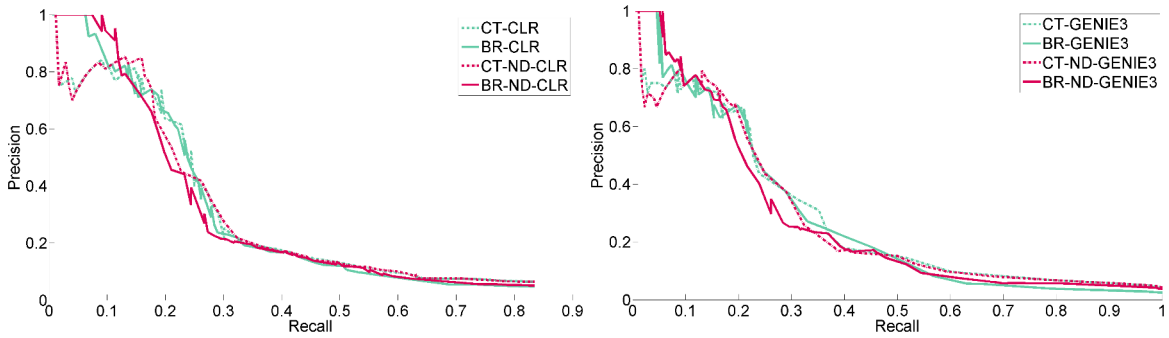
5.3 *BRANE Relax*: objective results on benchmark datasets

BRANE Relax performance is assessed through the methodology provided in Section 3.2. From each simulated dataset given by the DREAM4 (Marbach *et al.*, 2010) and DREAM5 (Marbach *et al.*, 2012) challenges, weights of the complete graph are obtained using either CLR (Faith *et al.*, 2007) or GENIE3 (Huynh-Thu *et al.*, 2010). We also carried out a comparative evaluation on CLR or GENIE3 weights improved by the post-processing Network Deconvolution (Feizi *et al.*, 2013). Each generated weighted complete graph is then gradually pruned, thanks to the classical thresholding (CT) or our approach *BRANE Relax*, by varying the λ parameter. This procedure allows us to compute a set of Precision (3.10) - p. 59 and Recall (3.11) - p. 59 values yielding Area Under Precision-Recall curves. Finally, this measure is used to compare *BRANE Relax* to CT, from CLR, ND-CLR, GENIE3 and ND-GENIE3 weights. Note that *BRANE Relax* formulation (5.12) involves a function Φ evaluating the current node degrees with respect to the fixed one d, set to 3 in advance from biological knowledge. We firstly present results with the intuitive squared ℓ_2 norm — $\Phi(\cdot) = \|\cdot\|^2$ —, and study, over a second phase, the impact of the choice of the function Φ on the results. We fixed the regularization parameter μ to 0.005 in all our simulations.

5.3.1 Numerical performance on DREAM4

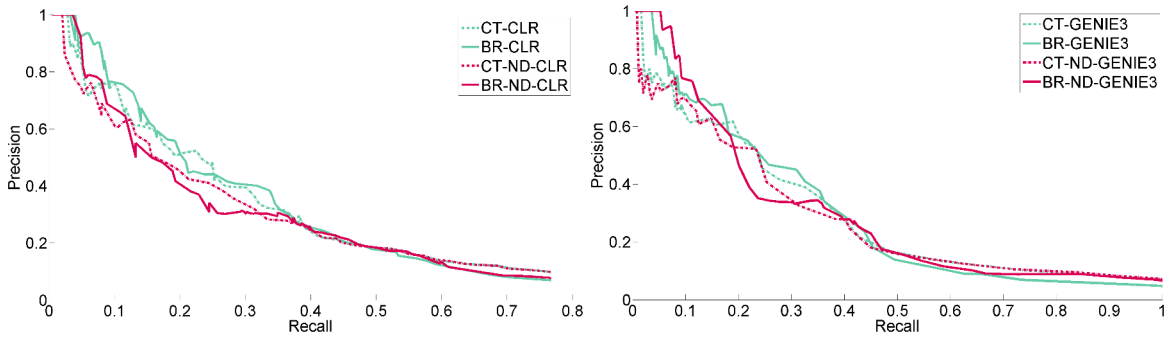
We first study results obtained on the five datasets of DREAM4. Precision-Recall (PR) curves are displayed in Figures 5.3 to 5.7. We recall that in such a curves, zones of higher importance is located on the top-left part as they corresponds to networks with relatively high Precision values in addition to have interpretable size i.e. networks with less than 1000 edges and having a precision greater than 50 %. If improvements are expected, they should be preferentially located in the top-left part of PR curves.

At first glance on all datasets and initial weights, PR curves obtained for *BRANE Relax* are above those obtained with CT. In addition, *BRANE Relax* curves show the anticipated effect previously mentioned as they exhibit significant improvements on the top-left part. As a complement to PR curves, numerical results, in terms of AUPRs and their relative gains, for the five datasets of DREAM4 are given in Table 5.1. Specifically in Table 5.1(a), first and second best performers are highlighted in italics and always refer to *BRANE Relax*. In addition, except



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.3 ~ PR CURVES FOR THE DATASET 1 OF DREAM4 (Q-*BRANE Relax*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the squared ℓ_2 norm for Φ, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

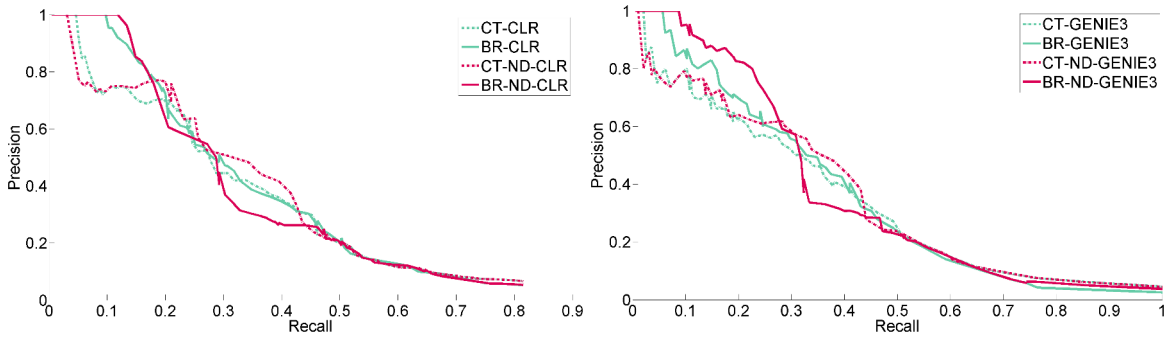


(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.4 ~ PR CURVES FOR THE DATASET 2 OF DREAM4 (Q-*BRANE Relax*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the squared ℓ_2 norm for Φ, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

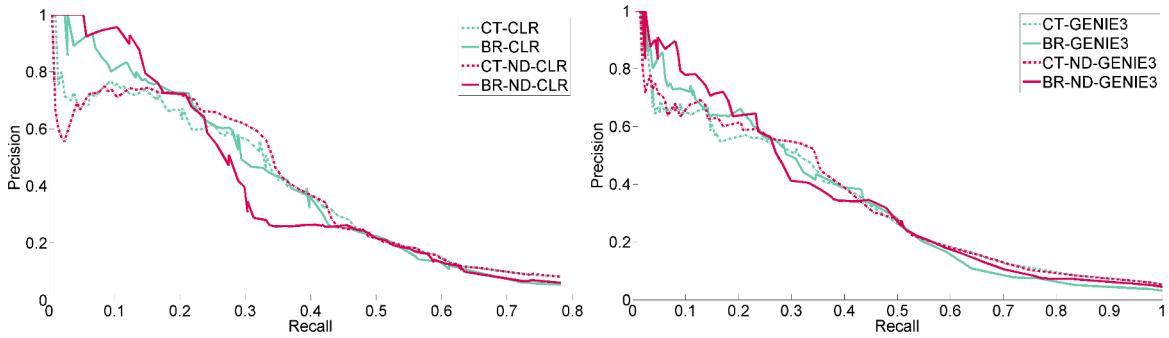
for two cases (on Network 4 and 5 with ND-CLR weights), each method tested (CLR, GENIE3, ND-CLR or ND-GENIE3) used as initialization exhibits an improved AUPR with *BRANE Relax* post-processing. While results on ND-CLR shows a null average gain over the five datasets, average gains reach 5.7 %, 3.2 % and 4.2 % on CLR, GENIE3 and ND-GENIE3, respectively (see Table 5.1(b)).

Despite the positive results we obtained on these datasets, improvements can appear weak as the maximal improvement is lower than 10 %. However, as it can be observed on the PR curves, differential improvements are observed across different parts of the curves. Focusing the assessment on areas of higher importance (top-left part), we can observe, for all the tested methods, a significant improvement of the results when *BRANE Relax* is used. Notably, these improvement can be illustrated through the capability to infer perfect networks (with a Precision value equal



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.5 ~ PR CURVES FOR THE DATASET 3 OF DREAM4 (Q-*BRANE Relax*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the squared ℓ_2 norm for Φ, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

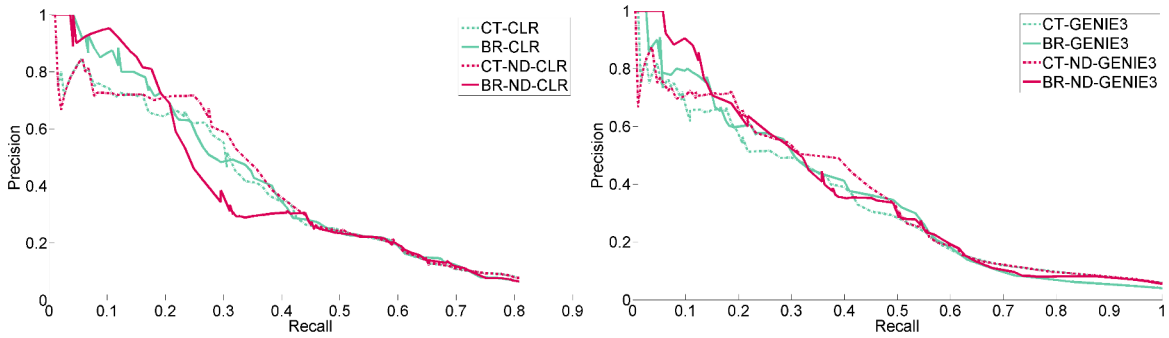


(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.6 ~ PR CURVES FOR THE DATASET 4 OF DREAM4 (Q-*BRANE Relax*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the squared ℓ_2 norm for Φ, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

to 1), corresponding to curve plateaus. The largest network obtained with CT with a maximal Precision value contains 9 edges. It is obtained on the third dataset with CLR weights. Using the same dataset and weights, *BRANE Relax* infer, at the maximal precision, an about twice larger network with ten additional edges. Conversely, the largest network obtained by *BRANE Relax* at the maximal precision over the five datasets and the four initial edge weights (CLR, GENIE3, ND-CLR and ND-GENIE3) contains 23 edges. More globally, perfectly inferred networks are, in average over the 5×4 = 20 studied cases, of size of 3 and 11 for CT and *BRANE Relax*, respectively and in average 4.8 times larger. These observations suggesting a more reliable inference process using *BRANE Relax* are valid for high Precision (larger than 85 %) as well.

For a complementary point of view, an evaluation of the post-processing itself can be considered. As provided in Table 5.2, the comparison of the AUPRs obtained with ND or *BRANE Relax*



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.7 ~ PR CURVES FOR THE DATASET 5 OF DREAM4 (Q-*BRANE Relax*) ~
Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the squared ℓ_2 norm for Φ, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.

Dataset	1	2	3	4	5	Average
CT-CLR	0.256	0.275	0.314	0.313	0.313	0.294
BR-CLR	*0.267*	0.282	0.337	*0.327*	0.344	0.311
CT-GENIE3	0.269	0.288	0.331	0.323	0.329	0.308
BR-GENIE3	*0.271*	*0.296*	*0.349*	*0.327*	*0.348*	*0.318*
CT-ND-CLR	0.254	0.250	0.324	0.318	0.331	0.295
BR-ND-CLR	0.255	0.252	0.324	0.317	0.328	0.295
CT-ND-GENIE3	0.263	0.275	0.336	0.328	0.354	0.309
BR-ND-GENIE3	0.264	*0.293*	*0.364*	*0.341*	*0.364*	*0.325*

(a) AUPRs.

Dataset	1	2	3	4	5	Average
BR-CLR *vs* CT-CLR	4.3 %	2.4 %	7.2 %	4.7 %	9.8 %	5.7 %
BR-GENIE3 *vs* CT-GENIE3	0.6 %	2.8 %	5.5 %	1.5 %	5.7 %	3.2 %
BR-ND-CLR *vs* CT-ND-CLR	0.2 %	0.8 %	0 %	−0.2 %	−0.8 %	0 %
BR-ND-GENIE3 *vs* CT-ND-GENIE3	0.3 %	6.4 %	8.1 %	3.7 %	2.7 %	4.2 %

(b) Relative gains.

Table 5.1 ~ NUMERICAL PERFORMANCE ON DREAM4 (*BRANE Relax*) ~
(a) Area Under PR curve (AUPR) obtained using CT or BRANE Relax (BC) on CLR, ND-CLR, GENIE3 and ND-GENIE3 weights. Weights are computed for each dataset (1 to 5) of the DREAM4 multifactorial challenge. Average AUPR are also reported as well as the two maximal improvements (in italics). (b) Relative gains obtained by comparing BRANE Relax to CT.

on CLR and GENIE3 weights is in favor of *BRANE Relax* with an average improvement reaching

5.8 % and 2.2 %, respectively. Analyzing detailed gains in Table 5.2, we observe two negative

Dataset	1	2	3	4	5	Average
BR-CLR *vs* CT-ND-CLR	5.1 %	12.8 %	4 %	2.8 %	4.2 %	5.8 %
BR-GENIE3 *vs* CT-ND-GENIE3	3.0 %	6.5 %	3.6 %	−0.6 %	−1.7 %	2.2 %

Table 5.2 ~ POST-PROCESSING PERFORMANCE ON DREAM4 (*BRANE Relax*) ~
Relative gains computed using AUPRs provided in Table 5.1(a) and are given for BRANE Relax using CLR (resp. GENIE3) weights compared to CT using ND-CLR (resp. ND-GENIE3).

gains for Network 4 and 5 using the GENIE3 weights. In addition to be lower than the smallest positive gain we obtained, these results are mainly due to some degradations which can occur in intermediate Precision and Recall. These ranges not being of highest importance in terms of biological interpretation, conclusions regarding these degradations can be balanced.

Before pursuing the assessment on a more realistic dataset, we further study the impact of the choice of the function Φ in the *BRANE Relax* formulation (5.12).

5.3.2 Impact of the function Φ

As mentioned, we firstly chose the function Φ in (5.12) as the squared ℓ_2 norm. However, this function is known to be sensitive to the outliers. In order to overcome this sensitivity, an $\ell_2 - \ell_1$ function can be considered. For this purpose, we also assess the performance of *BRANE Relax* using for Φ the Huber potential function (Huber, 1964), illustrated in Figure 5.8.

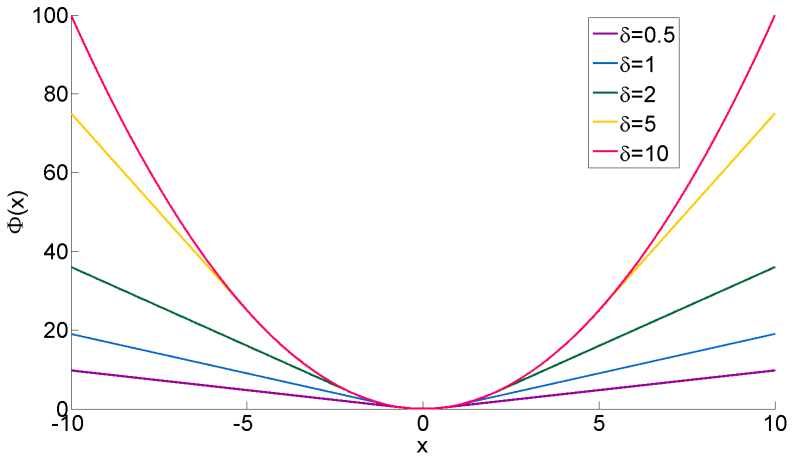


Figure 5.8 ~ HUBER FUNCTION FOR VARIOUS δ PARAMETERS ~

This loss function involves a parameter δ. Based on our formulation in (5.9), this loss function

is expressed as

$$\phi(y_i) = \begin{cases} y_i^2 & \text{if } |y_i| \leq \delta, \\ 2\delta(|y_i| - \frac{1}{2}\delta) & \text{otherwise,} \end{cases} \tag{5.26}$$

where y_i is the difference between the current degree of node i and the constant d i.e. for all $i \in \{1, \ldots, P\}, y_i = \sum_{k=1}^{E} \Omega_{i,k} x_k - d$. In the interval $[-\delta, \delta]$, the Huber function has a quadratic behavior while a linear one appear outside this interval. Note that, for sufficiently large δ values, the quadratic behavior is recovered for limited amplitude data. Due to its potential robust norm behavior, using the Huber function instead of the squared ℓ_2 norm could be a judicious choice. Indeed, as it is expected to be more robust to outliers for suitable δ parameter, the Huber function can appear useful, especially on real data.

AUPRs obtained with *BRANE Relax* using the Huber function (hBR) with a parameter δ fixed to 0.1 for all the simulations are provided in Table 5.3. For each tested dataset and initial weights, we also provide gains over either CT or *BRANE Relax* using a quadratic function for Φ. Note that corresponding PR curves are displayed at the end of this chapter in Figures 5.12 to 5.16.

	1	2	3	4	5	Average
hBR-CLR	0.278	0.293	0.336	0.333	0.345	0.317
hBR-CLR vs CT-CLR	8.6 %	6.7 %	6.9 %	6.4 %	10.2 %	7.8 %
hBR-CLR vs qBR-CLR	4.1 %	3.9 %	−0.3 %	1.8 %	0.3 %	2.0 %
hBR-GENIE3	0.293	0.320	0.356	0.345	0.354	0.334
hBR-GENIE3 vs CT-GENIE3	8.9 %	11.3 %	7.6 %	7.2 %	7.6 %	8.5 %
hBR-GENIE3 vs qBR-GENIE3	8.1 %	8.1 %	2.0 %	5.5 %	1.7 %	5.1 %
hBR-ND-CLR	0.270	0.264	0.327	0.325	0.332	0.304
hBR-ND-CLR vs CT-ND-CLR	6.4 %	5.7 %	0.9 %	2.2 %	0.3 %	3.1 %
hBR-ND-CLR vs qBR-ND-CLR	5.9 %	4.8 %	0.9 %	2.5 %	1.2 %	3.1 %
hBR-ND-GENIE3	0.276	0.307	0.369	0.347	0.371	0.334
hBR-ND-GENIE3 vs CT-ND-GENIE3	4.7 %	11.5 %	9.6 %	5.6 %	4.8 %	7.3 %
hBR-ND-GENIE3 vs qBR-ND-GENIE3	4.5 %	4.8 %	1.4 %	1.7 %	1.9 %	2.9 %

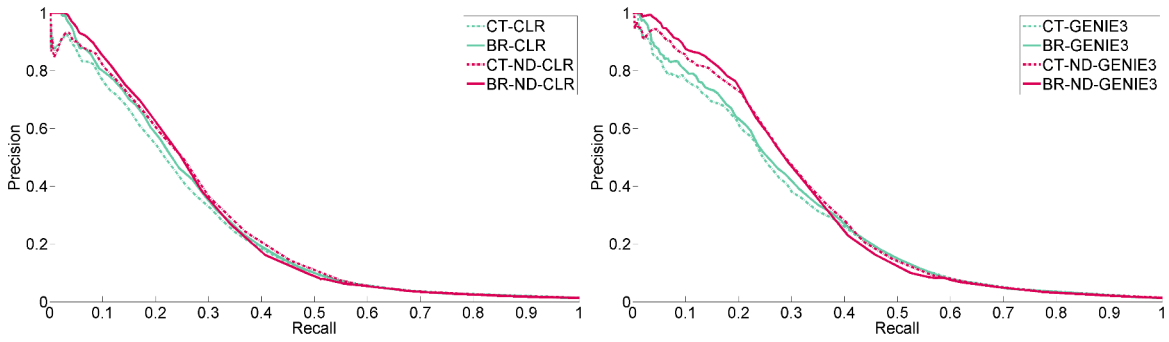
Table 5.3 ~ IMPACT OF THE FUNCTION Φ ON AUPRS ~
AUPRs correspond to BRANE Relax with the Huber function for Φ (hBR). Gains are given by comparing hBR to CT or BRANE Relax with the quadratic function for Φ (qBR).

As we can see in Table 5.3, *BRANE Relax* with the Huber function provides better results than with the squared ℓ_2 norm. In average, the maximal improvement reaches about 5 %. As a result, comparison to CT are thus even better with average gains equal to 7.8 %, 8.5 %, 3.1 % and 7.3 %, on CLR, GENIE3, ND-CLR and ND-GENIE3, respectively. In addition, significant improvement on the top-left part of PR curves are also recovered. These results show the

advantage of using the Huber function instead of the quadratic one for evaluating the current node degree with respect to the constant d. We thus carried an additional evaluation of *BRANE Relax* with the Huber function for Φ on the more realistic dataset from DREAM5.

5.3.3 Numerical performance on DREAM5

In the view of the satisfying validation on the five simulated dataset of DREAM4, we now present additional results on the simulated dataset provided by DREAM5. As mentioned, *BRANE Relax* was used with the Huber function for Φ with the parameter δ equal to 0.1 and the regularization parameter μ was set to 0.005. PR curves are displayed in Figure 5.9 and the associated AUPRs and relative gains are summarized in Table 5.4.



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.9 **~** PR CURVES FOR THE DATASET 1 OF DREAM5 (*BRANE Relax*) **~** *Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the Huber function for Φ, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

	AUPR	Gain		AUPR	Gain
CT-CLR	0.252	5.7 %	CT-ND-CLR	0.266	0.6 %
BR-CLR	0.272		BC-ND-CLR	0.274	
CT-GENIE3	0.283	3.8 %	CT-ND-GENIE3	0.313	0.3 %
BR-GENIE3	0.294		BC-ND-GENIE3	0.314	

Table 5.4 **~** NUMERICAL PERFORMANCE ON DREAM5 (*BRANE Relax*) **~** *Area Under Precision-Recall curve (AUPR) obtained using CT or BRANE Relax with Huber function Φ on CLR, ND-CLR, GENIE3 or ND-GENIE3 weights computed from dataset 1 of the DREAM5 challenge. Relative gains between CT and BRANE Relax are also reported.*

Results shown in Table 5.4 exhibit positive gains reaching 5.7 %, 3.8 %, 0.6 % and 0.3 %, on CLR, GENIE3, ND-CLR and ND-GENIE3, respectively. Note that in this more realistic dataset, improvements are more significant on CLR and GENIE3 weights than on their improved

version by ND. On the post-processed weights, *BRANE Relax* and CT provided similar results and become competitive. Results also demonstrate the capability of *BRANE Relax* to infer more biological relevant networks, with specifically improved networks located on the top-left part of PR curves.

5.3.4 Speed-up performance

BRANE Relax delivers good results on the simulated benchmark datasets provided by DREAM4 and DREAM5. Its convergence speed is also interesting, especially the reward reaped from the two acceleration tricks we used: preconditioning and block coordinate strategy. It is thus obvious to compare, in a first phase, the convergence speed of the accelerated version of *BRANE Relax* (P-FB and BC-P-FB) to the standard one (FB). In addition, convergence speed can also be compared to FISTA (Beck and Teboulle, 2009). A measure of convergence of our solution is provided by the variation of $(\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}\|/\|\hat{\boldsymbol{x}}\|)_{k\in\mathbb{N}}$, where $\boldsymbol{x}_k$ is the current edge labeling at iteration k of the algorithm and $\hat{\boldsymbol{x}}$ is the optimal solution, computed — in advance — over a large number of iterations. To give an idea about the computation times obtained in practice[1], our algorithm took about 15 seconds to infer a 155-edges network without acceleration tricks. The preconditioning reduces the computation time to 2 seconds and, by combining the block coordinate strategies to the previous one, the network is inferred in only 0.25 seconds. In comparison, FISTA took 6 seconds to solve the same optimization problem. Another graphical illustration of the speed gain for *BRANE Relax* implemented using standard Forward-Backward (FB), preconditioned FB (P-FB) and block-coordinate plus preconditioning FB (BC-P-FB) is given through convergence profiles in Figure 5.10, in addition to the one obtained using FISTA, at higher relative errors. Results are obtained on Network 1 of the DREAM4 challenge with CLR weights, using the squared ℓ_2 norm for the Φ function and the regularization parameter μ set to 0.005.

As expected, both preconditioning and block coordinate strategies improve the convergence speed of the forward-backward (FB) algorithm we used. In addition, while FISTA exhibits better convergence speed than FB, the complete version resulting in BC-P-FB appears largely faster. Additionally, for the BC-P-FB implementation, it could be interesting to study the impact of the number of blocks on the convergence speed. For this purpose, we vary the number of blocks and evaluate the stopping time with the same criterion as before ($\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}\|/\|\hat{\boldsymbol{x}}\| \leq 10^{-5}$). Note that this analysis was performed for the number of blocks giving equally-sized blocks only. The results presented in Figure 5.11 come from the Network 3 of the DREAM4 challenge using the GENIE3 weights. The Huber function was chosen with a parameter δ equal to 0.1 and a regularization parameter μ set to 0.005. As we can see in Figure 5.11, the best speed-up was found using $J = 3$.

5.4 Conclusions on *BRANE Relax*

BRANE Relax optimization is designed to perform an edge selection in a complete weighted network for GRN refinement. As *BRANE Cut*, it integrates biological *a priori* enforcing a modular

[1] Intel i7-3740QM @ 2.70GHz / 8 Gb RAM, Matlab 2011b.

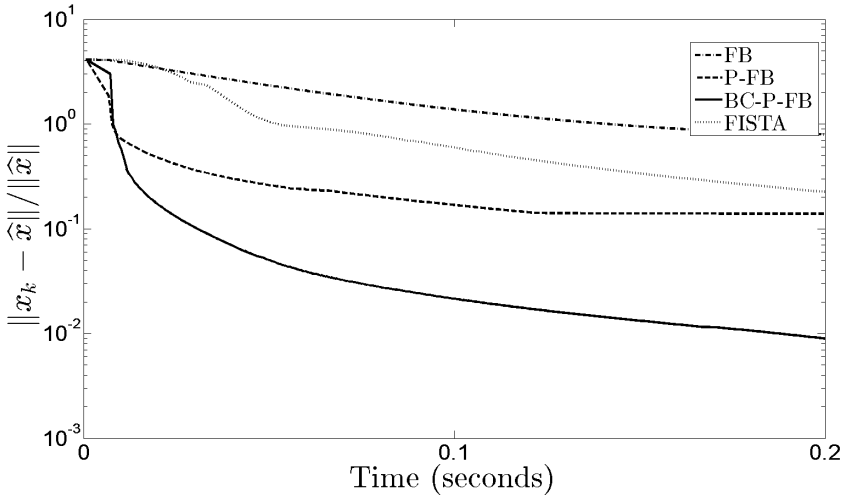


Figure 5.10 ~ Convergence profiles for various algorithms solving *BRANE Relax* ~

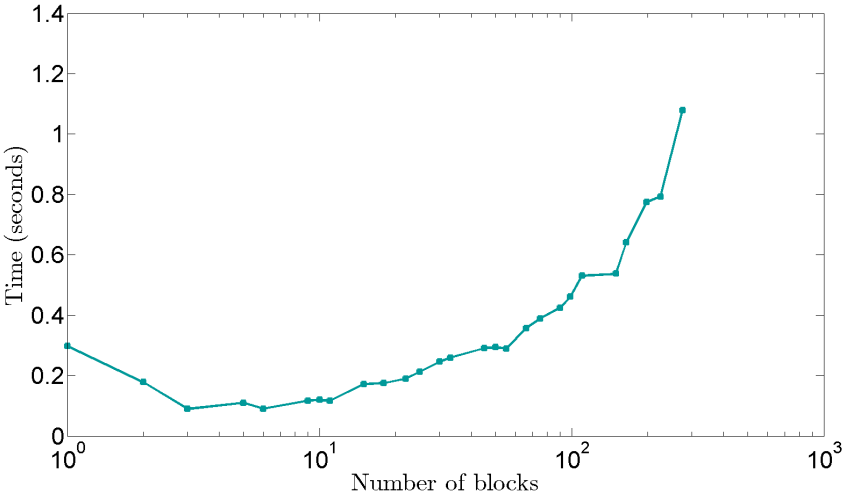
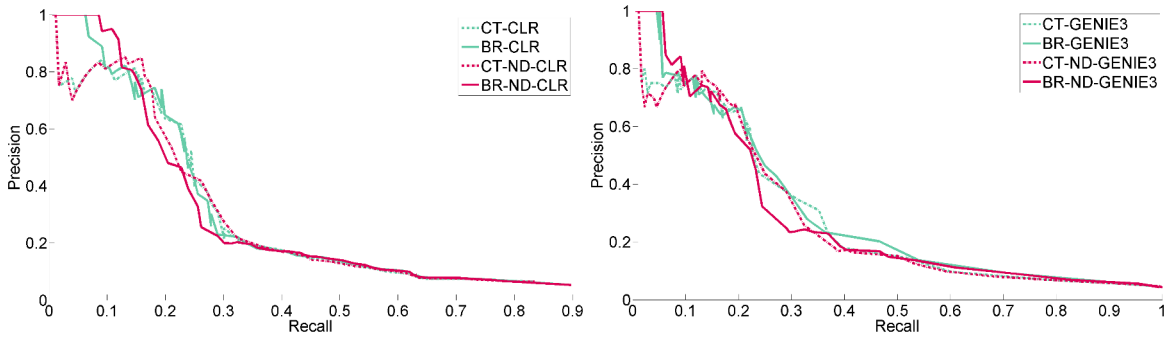


Figure 5.11 ~ Convergence time dependence on block size for BC-P-FB implementation of *BRANE Relax* ~

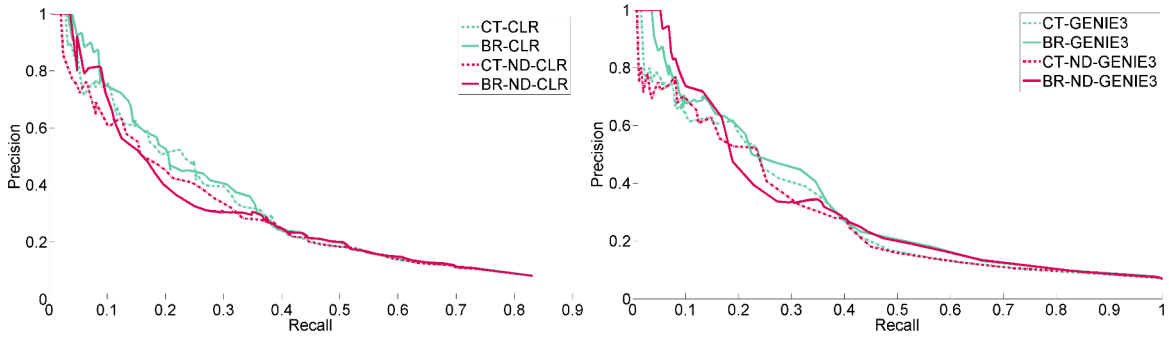
structure of the final network. It replaces co-regulation enforcement by a restriction of the connectivity degree of genes not identified to code for transcription factors. The latter was initially thought as a soft constraint, easier to fix with biological knowledge than co-regulation weights. The resulting optimization problem can be solved by a proximal splitting strategy yielding the use of an efficient variant of a projected gradient algorithm. In addition, preconditioning and block coordinate strategies are used to improve convergence speed. Its performance is demonstrated through the simulated datasets provided in the challenge DREAM4 and DREAM5 and

shows improvement over state-of-the-art methods. However, it finally slightly lagged behind *BRANE Cut*, after an additional work on improved weight initialization.



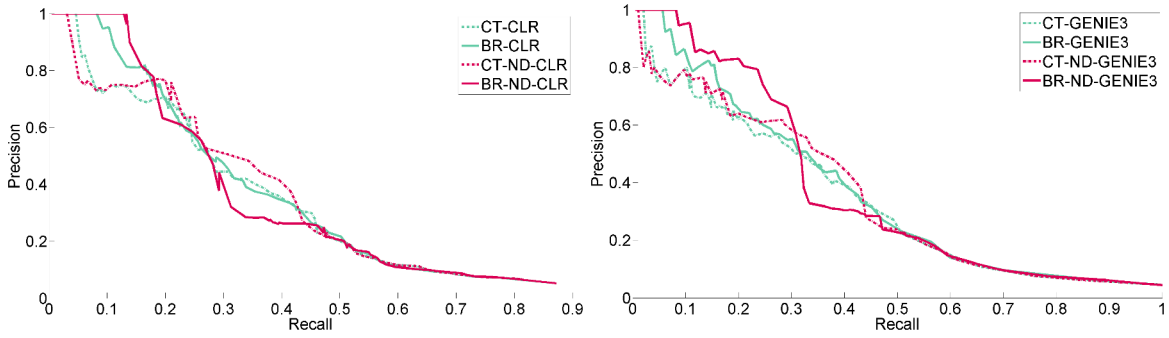
(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.12 ~ PR CURVES FOR THE DATASET 1 OF DREAM4 (H-*BRANE Relax*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the Huber function for* Φ*, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*



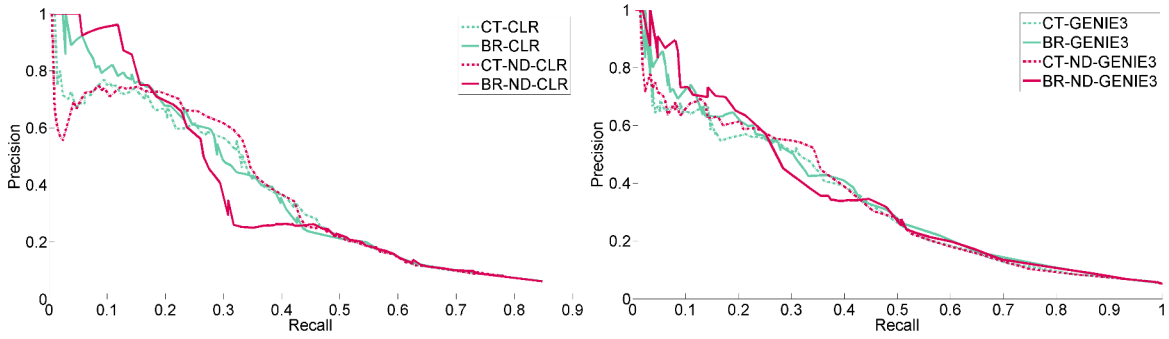
(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.13 ~ PR CURVES FOR THE DATASET 2 OF DREAM4 (H-*BRANE Relax*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the Huber function for* Φ*, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*



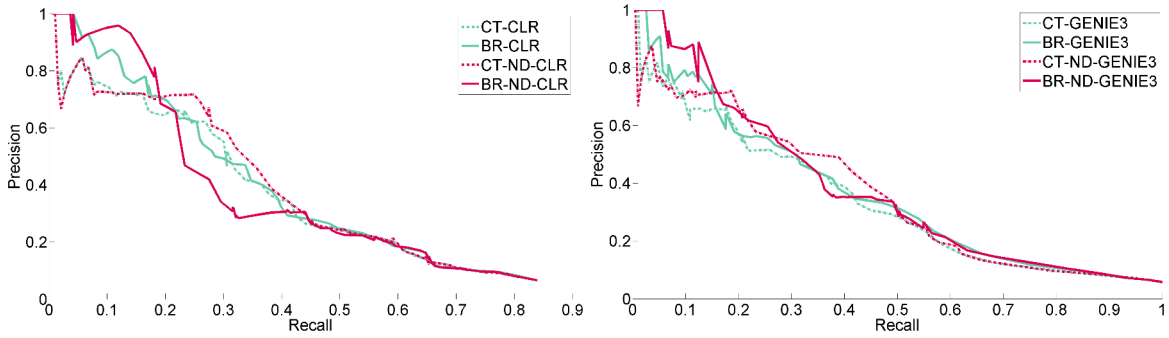
(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.14 **~** PR CURVES FOR THE DATASET 3 OF DREAM4 (H-*BRANE Relax*) **~** *Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the Huber function for* Φ*, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.15 **~** PR CURVES FOR THE DATASET 4 OF DREAM4 (H-*BRANE Relax*) **~** *Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the Huber function for* Φ*, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 5.16 ~ PR CURVES FOR THE DATASET 5 OF DREAM4 (H-*BRANE Relax*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Relax, with the Huber function for* Φ*, on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

6 Edge selection refinement using node clustering (*BRANE Clust*)

"My belief is that nothing that can be expressed by mathematics cannot be expressed by careful use of literary words."

Paul Samuelson

This chapter is dedicated to the detailed presentation of BRANE Clust for which a preliminary version is available in Pirayre et al. (2015c) and is extended in Pirayre et al. (2018a). In the same vein as our global methodology, BRANE Clust is designed to be applied to every complete graph for edge selection refinement in a GRN context. For this purpose, our formulation adapts graph weights by embedding clustering a priori. A modular graph structure is constrained through a TF*-centric semi-supervised clustering. The resulting cluster-assisted inference problem is solved* via *an alternating optimization scheme including the resolution of a linear system of equations involving the graph Laplacian matrix. Numerical results obtained on benchmark datasets from DREAM4 and DREAM5 are compared to state-of-the-art methods. The biological added value of BRANE Clust is also provided through a comparative analysis of* Escherichia coli *networks.*

Contents

6.1 Complemental works on joint clustering and inference

While we provide in Section 3.1 a relatively well-detailed overview of related works on GRN inference, we did not detail methods integrating clustering aspects. We thus provide some additional information here.

As mentioned many times, in a GRN context, finding genuine edges among all possible edges is still a challenging task especially due to the large number of genes with respect to the number of experiments. While we propose, in *BRANE Cut* and *BRANE Relax*, to restrict the space of possibilities by integrating biological-based constraints, one can benefit from the incorporation of modular structures at earlier stages of GRN inference. Notably, compounding inference and clustering more directly can better take network topology into account (Newman, 2012). Nevertheless, to the best of our knowledge, only very few methods integrate clustering information into graph inference task. They can be split into two classes according to the clustering usage.

On the one side, some of them use clustering task in an independent manner from the inference. In the works of Toh and Horimoto (2002) and Horimoto and Toh (2001), the clustering is firstly performed on gene expression data thanks to a hierarchical cluster analysis. Then, for each cluster, an average gene expression profile is computed. The dataset used for the inference thus corresponds to the average gene expression profile instead of all gene expression profiles. This procedure allows to decrease the number of genes/experiments ratio. A Gaussian Graphical Models (GGM) is then used on this new dataset to infer links between clusters yielding a reduced graph. Based on a complementary strategy, authors in Lee and Yang (2008) firstly perform a gene clustering via a SOM/SOFM procedure (self-organizing feature map) (Kohonen, 2000). Instead of inferring links between clusters, they infer links within each cluster using recurrent neural network approaches. As previously, inference is performed on reduced datasets for which the ratio between the number of genes and the number of experiments is more favorable. As a result, one network per cluster is obtained and an additional step is needed to aggregate results yielding to the final GRN. We note that, in WGCNA (Langfelder and Horvath, 2008), the clustering task is not used in pre-processing to help the inference. Conversely, it is performed *a posteriori*, by default *via* a hierarchical clustering, to detect gene modules from the correlation matrix encoding a GRN.

On the other side, clustering takes part in the inference. In Chiquet *et al.* (2009), authors used GGM to infer the GRN. Their formulation based on a maximum likelihood framework integrates a penalization on hidden clusters encoding a latent structure of the network. Both the latent structure and the concentration matrix are determined thought an alternating strategy relying on the EM (Expectation-Maximization) algorithm combining Bayes variational and Lasso-like procedures. In Roy *et al.* (2013), the authors use probabilistic graphical models integrating a prior taking into account co-regulation aspects of potential gene regulators to promote modular GRN. A clustering is firstly performed in order to initialize gene modules. Then, their algorithm identify regulators and infer modules in an alternating manner. In the same vein, an iterative module learning procedure, based on the Expectation-Maximization algorithm, is proposed by Segal *et al.* (2003) to deal with a probabilistic graphical model. This procedure is

improved in Joshi *et al.* (2009) using a set of possible statistical models.

We now detail the *BRANE Clust* model developed in this thesis for cluster-assisted inference refinement purpose. We first explain the preliminary work we performed (*BRANE Clust* with *hard*-clustering) before detailing its extension related to a more realistic biological assumption (*BRANE Clust* with *soft*-clustering).

6.2 *BRANE Clust* with *hard*-clustering

Relying on sound and informative gene clustering, one can better control a modular graph structure. We consider here TF-centric modules as groups of genes arranged around transcription factors. This additional knowledge is used for prediction, as TF-centric modules favor the detection of new target genes. In this preliminary work, referred to as *BRANE Clust* with *hard*-clustering, TF-centric modules are constructed through semi-supervised clustering where only one TF is associated to (only) one cluster.

6.2.1 Problem formulation

In order to construct a cluster-assisted inference model, we integrate a clustering step into the classical thresholding (CT) (3.23) - p. 66. It promotes the presence of edges linking nodes belonging to the same cluster. For this purpose, we want to design a cost function so as to impact weights in (3.23) as follows. If nodes v_i and v_j belong to:

- ⋆ the same cluster i.e. $y_i = y_j$, weights remain unchanged,
- ⋆ distinct clusters i.e. $y_i \neq y_j$, weights are reduced.

Let $y \in \mathbb{N}^G$ denote a node cluster labeling vector. Let $\mathbb{1}(\cdot)$ denote the characteristic function equal to 1 if its argument is verified and 0 otherwise. A parameter $\beta > 1$ is used to control the clustering influence. An instance of cost function satisfying the above weight modification, bearing analogies with a Potts model (Yu, 1982), used for instance in community detection in graphs (Fortunato, 2010), is:

$$f(y_i, y_j) = \frac{\beta - \mathbb{1}(y_i \neq y_j)}{\beta}. \tag{6.1}$$

If nodes belong to the same cluster, $f(y_i, y_j) = 1$ independently of β. If nodes belong to different clusters, $f(y_i, y_j)$ equals $\frac{\beta-1}{\beta}$ and may vary from 0 (for close-to-one βs) to 1 (for higher βs), thus emulating standard thresholding.

The novel optimization problem can thus be simply re-expressed as:

$$\underset{\substack{\boldsymbol{x}\in\{0,1\}^E,\\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} f(y_i, y_j)\omega_{i,j}x_{i,j} + \lambda(1 - x_{i,j}) \tag{6.2}$$

where both variables $\boldsymbol{x}$ (edge binary labeling) and $\boldsymbol{y}$ (node clustering labeling) have to be optimized. However, this formulation does not integrate any constraint on the clustering. As we want to promote TF-centric modules i.e. clusters constructed around TFs, each TF node is pre-labeled by a distinct cluster label such that for all $i \in \mathbb{T}$, the cluster label y_i of the TF node v_i is fixed to i. In addition to promoting a modular structure in the graph, this constraint avoids a trivial solution for the clustering. Hence, adding this constraint to (6.2), the novel problem can thus be formulated as:

$$\begin{aligned} &\underset{\substack{\boldsymbol{x}\in\{0,1\}^E,\\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta - \mathbb{1}(y_i \neq y_j)}{\beta}\omega_{i,j}x_{i,j} + \lambda(1 - x_{i,j}) \\ &\text{subject to } y_i = i,\ \forall i \in \mathbb{T}. \end{aligned} \tag{6.3}$$

How to interpret the hard version of BRANE Clust? From a clustering viewpoint, it aims at obtaining TF-centric clusters. For this purpose, TFs are pre-labeled such that each TF belong to a distinct cluster. Here, the number of clusters is set to T (corresponding to the number of TFs) in an *ad hoc* manner. It thus remains at assigning to the $\overline{\text{TF}}$s a label pertaining to the set of pre-labels. Then, this clustering will impact the graph structure by preventing edges to appear across different clusters. This discrimination is encoded in the multiplicative factor $\frac{\beta-1}{\beta}$. Similarly, the linkage of the i-th TF node (thus belonging to cluster i) to $\overline{\text{TF}}$s nodes sharing the same cluster are fostered. Figure 6.1 illustrates the clustering effect on the graph structure inference through a toy example.

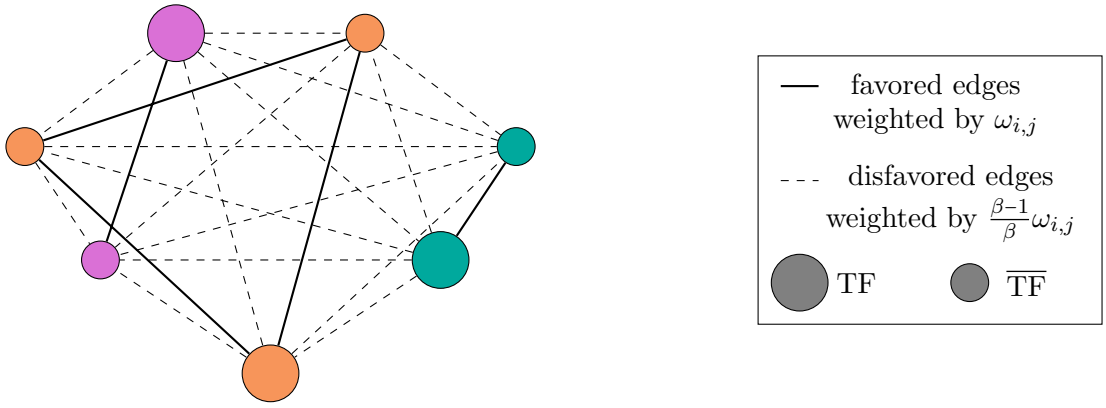


Figure 6.1 ~ *hard*-CLUSTERING EFFECT ON NETWORK INFERENCE ~
Large and smaller nodes correspond to TFs *and* $\overline{\text{TF}}$s*, respectively. Node colors encode cluster labels. This example is composed of 3* TFs *classified in 3 clusters (purple, orange and green).* $\overline{\text{TF}}$s *are assigned to one of the 3 clusters (light purple, orange and green). Links between nodes in the same cluster (solid lines) are favored while the others (dashed lines) are weakened.*

We now expose how the solution to the constrained optimization problem (6.3) is obtained.

6.2.2 Optimization framework

In (6.3), both the binary edge label $\boldsymbol{x}$ and the node label $\boldsymbol{y}$ have to optimized. Taking inspiration from a run of an alternating optimization scheme, *BRANE Clust* with *hard*-clustering can be solved in a one-shot procedure. First of all, let us consider Problem (6.3) at $\boldsymbol{y}$ fixed and $\boldsymbol{x}$ variable. In such a case, the optimal solution is explicit:

$$x^*_{i,j} = \begin{cases} 1 & \text{if } \omega_{i,j} > \frac{\lambda\beta}{\beta - \mathbb{1}(y_i \neq y_j)} \\ 0 & \text{otherwise,} \end{cases} \tag{6.4}$$

and can be expressed as:

$$x^*_{i,j} = \mathbb{1}\left(\omega_{i,j} > \frac{\lambda\beta}{\beta - 1}\right)\mathbb{1}(y_i \neq y_j) + \mathbb{1}(\omega_{i,j} > \lambda)\mathbb{1}(y_i = y_j). \tag{6.5}$$

From this result, we also find that

$$\begin{aligned} 1 - x^*_{i,j} &= \mathbb{1}\left(\omega_{i,j} \leq \frac{\lambda\beta}{\beta - 1}\right)\mathbb{1}(y_i \neq y_j) + \mathbb{1}(\omega_{i,j} \leq \lambda)\mathbb{1}(y_i = y_j), \\ &= \mathbb{1}\left(\omega_{i,j} \leq \frac{\lambda\beta}{\beta - 1}\right)\mathbb{1}(y_i \neq y_j) + \mathbb{1}(\omega_{i,j} \leq \lambda)(1 - \mathbb{1}(y_i \neq y_j)). \end{aligned} \tag{6.6}$$

Let us now consider Problem (6.3) at $\boldsymbol{x}$ fixed and optimal while $\boldsymbol{y}$ is variable, which is formulated as

$$\underset{\boldsymbol{y} \in \mathbb{N}^G \cap C}{\text{maximize}} \quad \sum_{(i,j) \in \mathbb{V}^2} \frac{\beta - \mathbb{1}(y_i \neq y_j)}{\beta} \omega_{i,j} x^*_{i,j} + \lambda(1 - x^*_{i,j}), \tag{6.7}$$

where

$$C = \left\{ (z_g)_{1 \leq g \leq G} \in \mathbb{R}^G \,|\, \forall i \in \mathbb{T}, z_i = i \right\} \tag{6.8}$$

encodes the pre-labeling constraint on TFs nodes. Equivalently, the problem can be re-expressed as

$$\underset{\boldsymbol{y} \in \mathbb{N}^G \cap C}{\text{maximize}} \quad \sum_{(i,j) \in \mathbb{V}^2} -\frac{\omega_{i,j}}{\beta} x^*_{i,j} \mathbb{1}(y_i \neq y_j) + (\lambda - \omega_{i,j})(1 - x^*_{i,j}). \tag{6.9}$$

By combining (6.5) and (6.6), we obtain:

$$\begin{aligned} \underset{\boldsymbol{y} \in \mathbb{N}^G \cap C}{\text{maximize}} \quad \sum_{(i,j) \in \mathbb{V}^2} & -\frac{\omega_{i,j}}{\beta} \mathbb{1}\left(\omega_{i,j} > \frac{\lambda\beta}{\beta - 1}\right)\mathbb{1}(y_i \neq y_j) \\ & + (\lambda - \omega_{i,j})\mathbb{1}(y_i \neq y_j)\left(\mathbb{1}\left(\omega_{i,j} \leq \frac{\lambda\beta}{\beta - 1}\right) - \mathbb{1}(\omega_{i,j} \leq \lambda)\right), \end{aligned} \tag{6.10}$$

that is

$$\underset{\boldsymbol{y} \in \mathbb{N}^G \cap C}{\text{maximize}} \quad \sum_{(i,j) \in \mathbb{V}^2} \mathbb{1}(y_i \neq y_j)\left[-\frac{\omega_{i,j}}{\beta}\mathbb{1}\left(\omega_{i,j} > \frac{\lambda\beta}{\beta - 1}\right) + (\lambda - \omega_{i,j})\left(\mathbb{1}\left(\omega_{i,j} \leq \frac{\lambda\beta}{\beta - 1}\right) - \mathbb{1}(\omega_{i,j} \leq \lambda)\right)\right] \tag{6.11}$$

Finally, optimization Problem (6.11) can be re-expressed into

$$\underset{\boldsymbol{y}\in\mathbb{N}^G\cap C}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \alpha_{i,j}\mathbb{1}(y_i \neq y_j), \tag{6.12}$$

where the weights $\alpha_{i,j}$ are given by

$$\begin{cases} 0 & \text{if } \omega_{i,j} < \lambda, \\ \omega_{i,j} - \lambda & \text{if } \lambda \leq \omega_{i,j} \leq \frac{\lambda\beta}{\beta-1}, \\ \frac{\omega_{i,j}}{\beta} & \text{if } \omega_{i,j} \geq \frac{\lambda\beta}{\beta-1}. \end{cases} \tag{6.13}$$

However, it turns out that Problem (6.12) is NP-hard (Darbon, 2009). In order to circumvent this difficulty, a continuous relaxation of this combinatorial problem can be introduced. To do so, assume that T is the number of clusters and introduce T vector variables $y^{(1)}, \ldots, y^{(T)}$ of size G, whose components are:

$$\forall i \in \mathbb{V} \quad \text{and} \quad \forall t \in \mathbb{T}, \quad y_i^{(t)} = \begin{cases} 1 & \text{if } y_i = t, \\ 0 & \text{otherwise.} \end{cases} \tag{6.14}$$

In addition to the relaxation, this decoupling strategy allows us to reformulate Problem (6.12) as follows:

$$\underset{\substack{y^{(1)}\in C^{(1)},\ldots,y^{(T)}\in C^{(T)} \\ (y^{(1)},\ldots,y^{(T)})\in D}}{\text{minimize}} \sum_{t=1}^{T}\Big(\sum_{(i,j)\in\mathbb{V}^2} \alpha_{i,j}(y_i^{(t)} - y_j^{(t)})^2\Big), \tag{6.15}$$

where, for every $t \in \{1, \ldots, T\}$,

$$C^{(t)} = \left\{ \left(z_g^{(t)}\right)_{1\leq g\leq G} \in \mathbb{R}^G \mid \forall i \in \mathbb{T}, z_i^{(t)} = s_i^{(t)} \right\}. \tag{6.16}$$

The vector $s^{(t)}$ — encoding the pre-labeling constraint on TFs — is defined from $t \in \mathbb{T}$ by a relation similar to (6.14), and

$$D = \left\{ \left(y^{(1)}, \ldots, y^{(T)}\right) \in \left(\{0,1\}^G\right)^T \mid \sum_{t=1}^{T} y^{(t)} = 1_G \right\}, \tag{6.17}$$

with $1_G = (1, \ldots, 1)^\top \in \mathbb{R}^G$. A convex relaxation of Problem (6.15) is then obtained by replacing D by its convex hull $\widehat{D}$

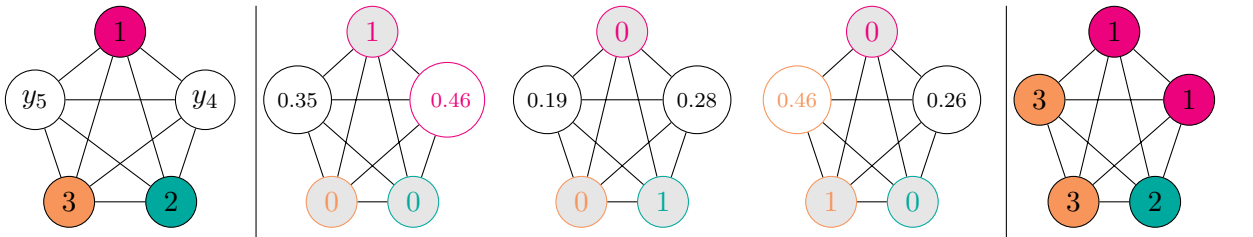
$$\widehat{D} = \left\{ \left(y^{(1)}, \ldots, y^{(T)}\right) \in \left([0,1]^G\right)^T \mid \sum_{t=1}^{T} y^{(t)} = 1_G \right\}. \tag{6.18}$$

In such a case, for all $t \in \mathbb{T}$, the vector $y^{(t)} \in [0,1]^G$ contains the probabilities for nodes to be assigned to cluster t. Provided that there is at least one pre-labeled node in each connected component of the graph, each of the T quadratic convex problems, known as the combinatorial Dirichlet problem, has a unique solution which can be obtained by solving a linear system of

equations. In addition, since the probabilities at each node will sum to unity, $T-1$ linear systems only need to be solved (Grady, 2006), as detailed in Section 3.3.3. Then, the final clustering label variable $y^* = (y_i^*)_{1\leq i\leq G}$ is given by

$$\forall i \in \mathbb{V}, \quad y_i^* = \arg\max_{t\in\mathbb{T}} \; y_i^{(t)}. \tag{6.19}$$

The proposed optimization problem — which can be assimilated to a random walker (Grady, 2006) — can be interpreted through a graph structure as illustrated in Figure 6.2. Indeed, for each sub-problem t, the graph interpretation resorts to fixing the marker label t (the t-th TF node) to 1 and the others to 0. Probability $y_i^{(t)}$ reflects the chance to reach the marker labeled by 1 first, for a random walker leaving node i in the graph. Higher weights encode preferable paths for the walker, and therefore drive the computed probabilities.



(a) initial network. (b) decoupling strategy. (c) final clustering.

Figure 6.2 ~ GRAPH INTERPRETATION FOR *BRANE Clust* WITH *hard*-CLUSTERING. ~
In the initial graph, colored nodes are TFs *and play the role of markers with fixed node label. It remains to assign a label to the* $\overline{\text{TF}}$ *nodes. Edge weights are given by* $\alpha_{i,j}$ *(6.13). The* T*-label problem is decoupled into* T *binary sub-problem. For each sub-problem* t*, the label of the corresponding marker is set to one and the others to zero. Probabilities for each* $\overline{\text{TF}}$ *nodes are then computed. The final node clustering corresponds to the label whose probability amidst the* T *sub-problems is maximal.*

Finally, the optimal clustering $\boldsymbol{y}^*$ is inserted in (6.5) to obtain the final edge labeling $\boldsymbol{x}^*$ yielding the final GRN. Altogether, our *BRANE Clust* algorithm with *hard*-clustering (Algorithm 2) can be summed up as follows:

We only provide preliminary results on simulated datasets before discussing possible improvements yielding an extended version of *BRANE Clust* with *soft*-clustering in Section 6.3.

6.2.3 Objective results

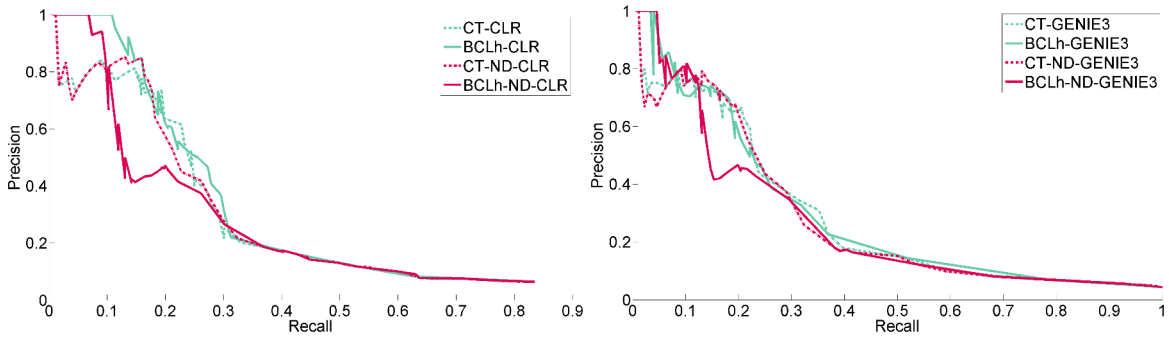
Based on the same methodology as previously, *BRANE Clust* with *hard*-clustering was preliminary evaluated on the five simulated datasets from DREAM4. For each dataset, a weighted complete

Algorithm 2: *BRANE Clust* with *hard*-clustering

Fix $\beta > 1$ and $\lambda \in [0, 1]$;

- Compute $\alpha_{i,j}$ weights using (6.13);
- Based on $\alpha_{i,j}$, compute the node label assignment probabilities $\mathcal{Y}$ solving the relaxed version of (6.15) with (6.18);
- Determine the optimal node cluster labeling $\boldsymbol{y}^\star$ with (6.19);
- Using $\boldsymbol{y}^\star$, compute the optimal labeling $\boldsymbol{x}^\star$ given by (6.4).

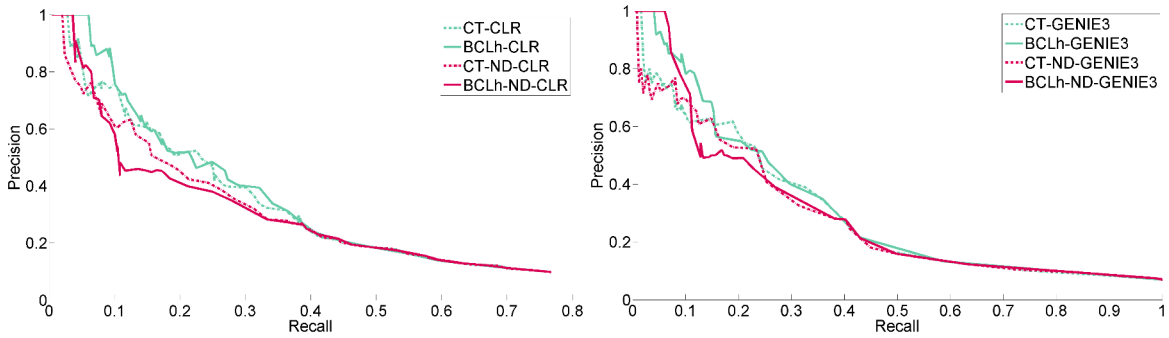
graph to be pruned is built using either CLR (Faith *et al.*, 2007) or GENIE3 (Huynh-Thu *et al.*, 2010). From this complete graph, a set of GRNs are obtained — by varying the threshold λ — for both classical thresholding (CT) and our approach *BRANE Clust*, yielding PR curves. AUPR for CT and *BRANE Clust* on CLR and GENIE3 weights are then computed and compared. Note that all *BRANE Clust* simulations are performed with the β parameter fixed to 2. Resulting PR curves are displayed in Figures 6.3 to 6.7 while Table 6.1 summarizes numerical performance in terms of AUPRs and relative gains.



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

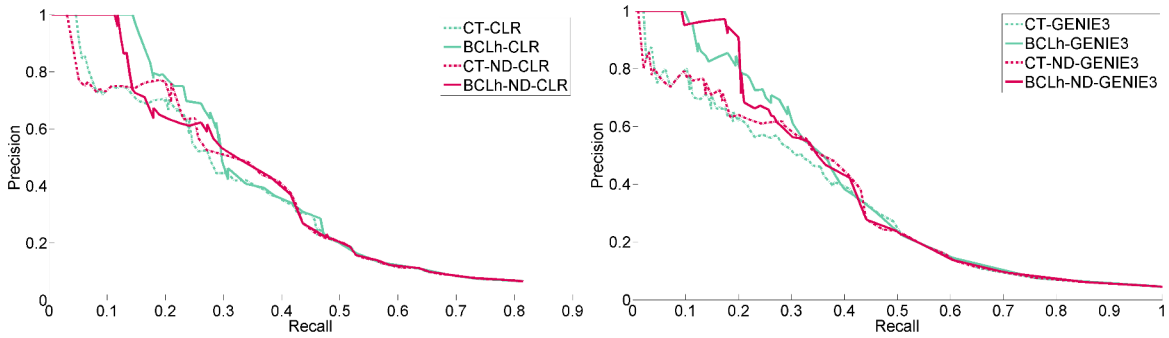
Figure 6.3 ~ PR CURVES FOR THE DATASET 1 OF DREAM4 (*BRANE Clust–hard*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* hard*-clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

From a global viewpoint on all tested datasets and initial weights (CLR, ND-CLR, GENIE3 or ND-GENIE3), *BRANE Clust* PR curves generally stand above CT PR curves. From Table 6.1(b), we can note an exception for three ND cases, for which relative gains are negatives, especially due to a degradation observed for intermediate precision values. Nevertheless, the average improvements reach 12 %, 11 %, 4.2 % and 7.5 % on the CLR, GENIE3, ND-CLR and ND-GENIE3 weights, respectively. In addition, as highlighted by the AUPRs in italics in Table 6.1, first and second best performances are always produced with *BRANE Clust*. We also



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.4 ~ PR CURVES FOR THE DATASET 2 OF DREAM4 (*BRANE Clust–hard*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* hard*-clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

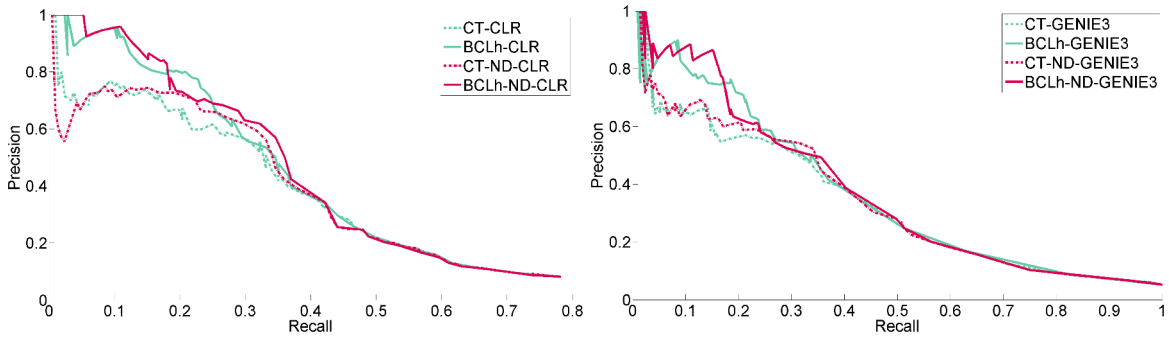


(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.5 ~ PR CURVES FOR THE DATASET 3 OF DREAM4 (*BRANE Clust–hard*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* hard*-clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

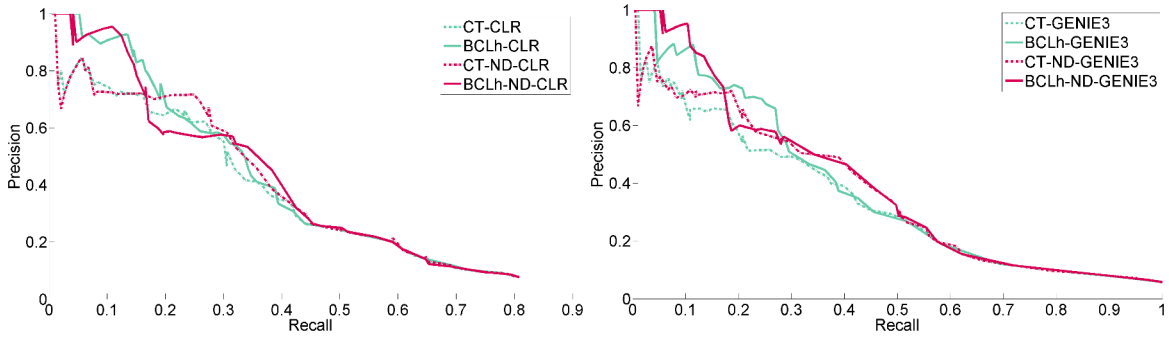
show that the most significant improvement is located in high-precision areas, for which networks are biologically relevant and interpretable. These results on DREAM4 suggest the judiciousness of the integration of a TF-centric clustering *a priori* during the inference step.

We also provide post-processing performance comparisons by comparing CT AUPRs on weights improved by ND to *BRANE Clust* AUPR on the non improved weights. Resulting relative gains are provided in Table 6.2. They show that *BRANE Clust* is a better post-processing method than ND with a minimal improvement of 4.2 % (obtained on dataset 5 and GENIE3 weights) and a maximal one reaching 15.2 % (obtained on dataset 2 and CLR weights). In average, *BRANE Clust* post-processing obtains an improvement of 11.9 % and 10.3 % on CLR and GENIE3, respectively, compared to ND post-processing. From a complete weighted adjacency matrix, it thus could be recommended to directly used *BRANE Clust* instead of a classical thresholding on



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.6 ~ PR CURVES FOR THE DATASET 4 OF DREAM4 (*BRANE Clust–hard*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* hard*-clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.7 ~ PR CURVES FOR THE DATASET 5 OF DREAM4 (*BRANE Clust–hard*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* hard*-clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

the weights improved by ND.

In a nutshell, preliminary results on DREAM4 are promising. In only three cases (out of twenty), slightly negatives gains are observed with respect to ND (−4.1 %, −1.6 % and −1.1 %). Their magnitude is smaller than all the other positive gains. Additionally, the degradation is mainly observed in areas on less importance.

We now assess *BRANE Clust* with *hard*-clustering on the simulated dataset 1 of the DREAM5 challenge. As previously, our approach is compared to the classical thresholding on initial weights obtained either with CLR or GENIE3. Post-processed CLR and GENIE3 weights with ND are also used. Resulting PR curves — obtained by varying the threshold parameter λ — are displayed in Figure 6.8 and the corresponding AUPR and gains are reported in Table 6.3. We note

Dataset	1	2	3	4	5	Average
CT-CLR	0.256	0.275	0.314	0.313	0.313	0.294
BCLh-CLR	*0.291*	0.288	0.358	0.356	0.355	0.330
CT-GENIE3	0.269	0.288	0.331	0.323	0.329	0.308
BCLh-GENIE3	*0.286*	*0.313*	*0.386*	0.360	*0.369*	*0.342*
CT-ND-CLR	0.254	0.250	0.324	0.318	0.331	0.295
BCLh-ND-CLR	0.244	0.247	0.342	*0.364*	0.352	0.310
CT-ND-GENIE3	0.263	0.275	0.336	0.328	0.354	0.309
BCLh-ND-GENIE3	0.259	*0.291*	*0.386*	*0.365*	*0.381*	*0.336*

(a) AUPRs.

Dataset	1	2	3	4	5	Average
BCLh-CLR *vs* CT-CLR	13.7 %	4.9 %	14.0 %	13.9 %	13.4 %	12.0 %
BCLh-GENIE3 *vs* CT-GENIE3	6.0 %	8.7 %	16.5 %	11.4 %	12.3 %	11.0 %
BCLh-ND-CLR *vs* CT-ND-CLR	−4.1 %	−1.1 %	5.5 %	14.5 %	6.3 %	4.2 %
BCLh-ND-GENIE3 *vs* CT-ND-GENIE3	−1.6 %	5.8 %	14.7 %	11.2 %	7.5 %	7.5 %

(b) Relative gains.

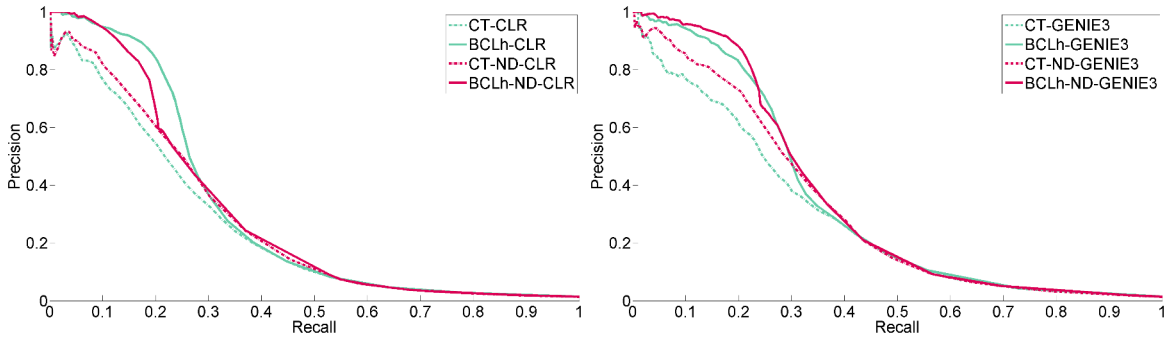
Table 6.1 ~ NUMERICAL PERFORMANCE ON DREAM4 (*BRANE Clust–hard*) ~
(a) Area Under PR curve (AUPR) obtained using CT or BRANE Clust with hard*-clustering (BCLh) on CLR, ND-CLR, GENIE3 and ND-GENIE3 weights. Weights are computed for each dataset (1 to 5) of the DREAM4 multifactorial challenge. Average AUPRs are also reported as well as the two maximal improvements (in italic). (b) Relative gains obtained by comparing BRANE Clust with* hard*-clustering to CT.*

that, as for the previous simulations, the parameter β controlling the influence of the clustering *a priori* is set to 2. Although (slightly) better outcomes could be observed with fine-tuning, we prioritized simplicity in comparisons, to set the ground for analyses where the ground truth is unknown.

As for previous results, *BRANE Clust* with *hard*-clustering shows refined results compared to CT, with a maximal improvement reaching about 22 % while the minimal improvement does not fall below 9 %. In addition, as observed in the PR curves of Figure 6.8, improvements are located on the top-left of the PR curves. While networks with a Precision higher than 80 % do not exceed a Recall of about 0.15 with CT, *BRANE Clust* allows to reach a Recall of about 0.25. This result suggests that for a given (and sufficiently high) Precision, *BRANE Clust* is able to detect more accurate graphs, thus containing more information. Finally, post-processing performance comparisons are also satisfactory. Indeed, comparing *BRANE Clust* with *hard*-clustering on initial CLR and GENIE3 weights with CT on the improved weights by ND yields gains reaching 13.2 % and 7.3 %. As a result, satisfactory numerical results are also obtained on this more realistic — although simulated — data.

Dataset	1	2	3	4	5	Average
BCLh-CLR *vs* CT-ND-CLR	14.6 %	15.2 %	10.5 %	11.9 %	7.2 %	11.9 %
BCLh-GENIE3 *vs* CT-ND-GENIE3	8.7 %	13.8 %	14.9 %	9.7 %	4.2 %	10.3 %

Table 6.2 ~ POST-PROCESSING PERFORMANCE ON DREAM4 (*BRANE Clust–hard*) ~
Relative gains computed using AUPRs provided in Table 6.1(a) are given for BRANE Clust with hard*-clustering using CLR (resp. GENIE3) weights compared to CT using ND-CLR (resp. ND-GENIE3).*



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.8 ~ PR CURVES FOR THE DATASET 1 OF DREAM5 (*BRANE Clust–hard*) ~
Precision-Recall (PR) curves obtained using CT or BRANE Clust with hard*-clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

	AUPR	Gain		AUPR	Gain
CT-CLR	0.252	22.2 %	CT-ND-CLR	0.272	9.2 %
BCLh-CLR	0.308		BCLh-ND-CLR	0.297	
CT-GENIE3	0.283	18.7 %	CT-ND-GENIE3	0.313	9.9 %
BCLh-GENIE3	0.336		BCLh-ND-GENIE3	0.344	

Table 6.3 ~ NUMERICAL PERFORMANCE ON THE DATASET 1 OF DREAM5 (*BRANE Clust–hard*) ~
Area Under Precision-Recall curve (AUPR) obtained using CT or BRANE Clust with hard*-clustering on CLR, ND-CLR, GENIE3 or ND-GENIE3 weights computed from dataset 1 of the DREAM5 challenge. Relative gains between CT and BRANE Clust are also reported.*

Notwithstanding, despite good performance obtained using *BRANE Clust* with *hard*-clustering, a non negligible limitation — inherent to the used *a priori* and invisible through the PR curves — can occur. Indeed, in high-precision networks, inferred modules can appear highly disconnected and, in such a case, biological relationships cannot be interpreted between modules. Hence, the

restriction to only one TF per cluster can be prejudicial in a real data context. *BRANE Clust* with *soft*-clustering has been developed to overcome this limitation by allowing cluster merging, authorizing multiple TFs in the same cluster.

6.3 *BRANE Clust* with *soft*-clustering

In this section, we adapt and extend *BRANE Clust* with *hard*-clustering to allow multiple TFs in the same cluster. The novel model, whose details are given in the following, is thus refereed to as *BRANE Clust* with *soft*-clustering.

6.3.1 Problem formulation

The *BRANE Clust* with *hard*-clustering model can be softened to better mimic biological scenarios. Indeed, TFs are expected to act in coordination suggesting — for our clustering *a priori* — the presence of several TFs in a same module. For this purpose, we propose to extend (6.3) to allow cluster merging instead of constraining only one TF per cluster. A possible generalization is

$$\underset{\substack{\boldsymbol{x}\in\{0,1\}^E,\\ \boldsymbol{y}\in\mathbb{N}^G}}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta - \mathbb{1}(y_i \neq y_j)}{\beta}\omega_{i,j}x_{i,j} \;+\; \lambda(1-x_{i,j}) \;+\; \sum_{i\in\mathbb{V},\, j\in\mathbb{T}} \mu_{i,j}\mathbb{1}(y_i = j), \tag{6.20}$$

where $\mu_{i,j}$ are weights controlling cluster merging. The third term of our model integrates the pre-labeled constraint. Indeed, in the characteristic function $\mathbb{1}(y_i \neq j)$, defined for all $i \in \mathbb{V}$ and all $j \in \mathbb{T}$, the node cluster label y_i is constrained to belong to $\mathbb{T}$ through a marker labeled by j. In such a case, the solution of the *hard*-clustering (6.3) can be recovered by setting

$$\mu_{i,j} = \begin{cases} \to \infty & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{6.21}$$

For *soft*-clustering, cluster fusion is driven by the $\mu_{i,j}$ weights. First of all, each TF $i \in \mathbb{T}$ is enforced exactly to be labeled by its native cluster i. This constraint is encoded through a $\mu_{i,j}$ equal to α when $i = j$, with $\alpha > 0$ chosen sufficiently high. In a second time, when $i \neq j$, cluster fusion has to be judiciously promoted. A simple merging criterion can result in detecting strong-enough relations between TFs and $\overline{\text{TF}}$s. For this purpose, a level $\tau \in [0,1]$ conditions the merging criterion defined by $\mathbb{1}(\omega_{i,j} > \tau)$. This criterion is weighted differentially with the nature of gene i. Indeed, when $i \in \mathbb{T}$, a large α factor allows node cluster label of TF i to be equal to j. In other words, if the edge linking TFs i and j has a weight $\omega_{i,j}$ higher than τ, TFs i and j have — without taking into account neighbors — the same chance to be assigned to the same cluster. Now, when $i \notin \mathbb{T}$, the merging criterion is weighted by $\omega_{i,j}$. This additional case allows us to preserve an influence of potentially undiscovered TFs. Consequently, we set:

$$\mu_{i,j} = \begin{cases} \alpha & \text{if } i = j\,, \\ \alpha\,\mathbb{1}(\omega_{i,j} > \tau) & \text{if } i \neq j \text{ and } i \in \mathbb{T}\,, \\ \omega_{ij}\,\mathbb{1}(\omega_{i,j} > \tau) & \text{if } i \neq j \text{ and } i \notin \mathbb{T}\,. \end{cases} \tag{6.22}$$

As introduced before, the α parameter controls the importance granted to the merge. When α is high, the merge is strongly promoted. Intuitively, cluster merging depends on strong-enough TF-TF relations. Indeed, the more the criterion merging is satisfied for TF-TF relationships (related to the proportion of weights above the threshold), the more the merge is promoted. We thus subsequently fix α to their cardinality:

$$\alpha = \sum_{(i,j)\in\mathbb{T}^2} \mathbb{1}(\omega_{i,j} > \tau). \tag{6.23}$$

This setting is consistent with the order of magnitude of optimal parameters obtained experimentally. Wrapping it up, *BRANE Clust* with *soft*-clustering allows cluster merging and thus multiple TFs in the same cluster. From an inference viewpoint, as edges linking nodes in the same cluster are preferably selected to the detriment of cluster-crossing edges, both combinatorial regulation and co-regulation could be promoted (Figure 4.2). The influence of the *soft*-clustering in the network inference is illustrated in Figure 6.9.

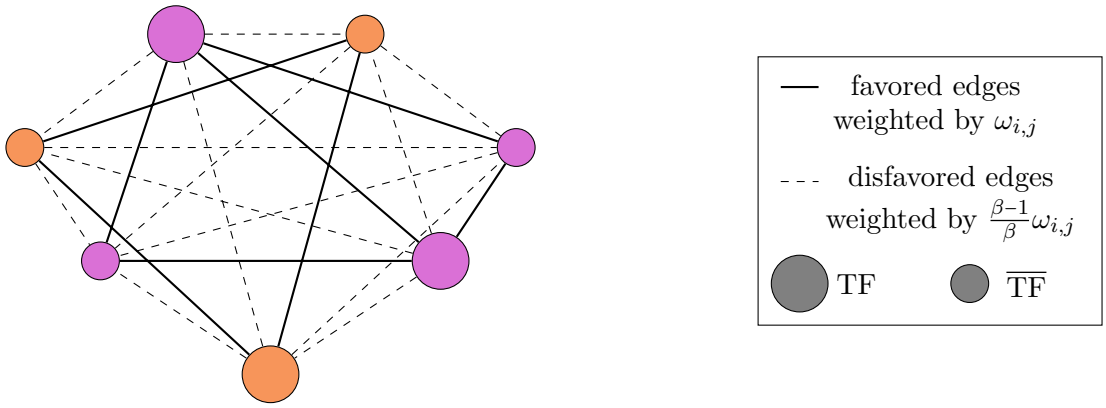


Figure 6.9 ~ *soft*-CLUSTERING EFFECT ON NETWORK INFERENCE ~
Large and smaller nodes correspond to TFs *and* $\overline{\text{TF}}$s*, respectively. Node color encodes cluster labels. This example is composed of 3 TFs classified in 2 clusters (purple and orange) thanks to the cluster merging capability of BRANE Clust.* $\overline{\text{TF}}$s *are assigned to one of the 2 clusters (light purple and light orange). Links between nodes in the same cluster (solid lines) are favored while the others (dashed lines) are depreciated.*

The proposed generalization now offers an inference formulation assisted by a clustering *a priori* with merging capability. We thus present the proposed procedure used to solve (6.20).

6.3.2 Optimization framework: alternating clustering and inference

From now on, we refer to *BRANE Clust* for this generalization, as we recall that Problem (6.20) encompasses both *hard* and *soft*-clustering according to the setting of weights $\mu_{i,j}$. In *BRANE Clust*, the optimization problem involves two kinds of variables: binary edge labeling $\boldsymbol{x}$ and node cluster labeling $\boldsymbol{y}$. It can thus be split into two sub-problems. *BRANE Clust* is then solved through

an alternating optimization scheme. At fixed $\boldsymbol{y}$ and variable $\boldsymbol{x}$, Problem (6.20) becomes:

$$\underset{\boldsymbol{x}\in\{0,1\}^E}{\text{maximize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\beta - \mathbb{1}(y_i \neq y_j)}{\beta}\omega_{i,j}x_{i,j} \quad + \quad \lambda(1 - x_{i,j})\,. \tag{6.24}$$

Its solution is explicit and is given by (6.4), as it is the case in the *hard*-clustering version of *BRANE Clust*. In such a case, we directly observe the influence of the clustering *a priori* on the inference. Indeed, if nodes v_i and v_j are in the same cluster, $y_i = y_j$ and the edge label $x_{i,j}$ will be equal to 1 if $\omega_{i,j} > \lambda$, as in the classical thresholding. Conversely, if nodes v_i and v_j are in distinct clusters, the optimal edge label $x_{i,j}$ will be 1 if the edge weight $\omega_{i,j}$ is higher than the new threshold defined by $\frac{\lambda\beta}{\beta-1}$. As we recall that $\beta > 1$, the new threshold is augmented, thus preventing edges crossing distinct clusters.

At fixed $\boldsymbol{x}$ and variable $\boldsymbol{y}$, Problem (6.20) reduces to

$$\underset{y\in\mathbb{N}^G}{\text{minimize}} \quad \sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}x_{i,j}}{\beta}\mathbb{1}(y_i \neq y_j) + \sum_{i\in\mathbb{V},\, j\in\mathbb{T}} \mu_{i,j}\mathbb{1}(y_i \neq j)\,. \tag{6.25}$$

Unfortunately, the cost function in (6.25) is NP-hard. In the same vein as (6.12), it can be harnessed with the random walker algorithm (Grady, 2006). Cluster labels are obtained by exactly relaxing simpler binary sub-problems. Binary label values relaxed in $[0,1]$ are interpreted as probabilities. Maximally probable outcomes finally yield optimal cluster labeling.

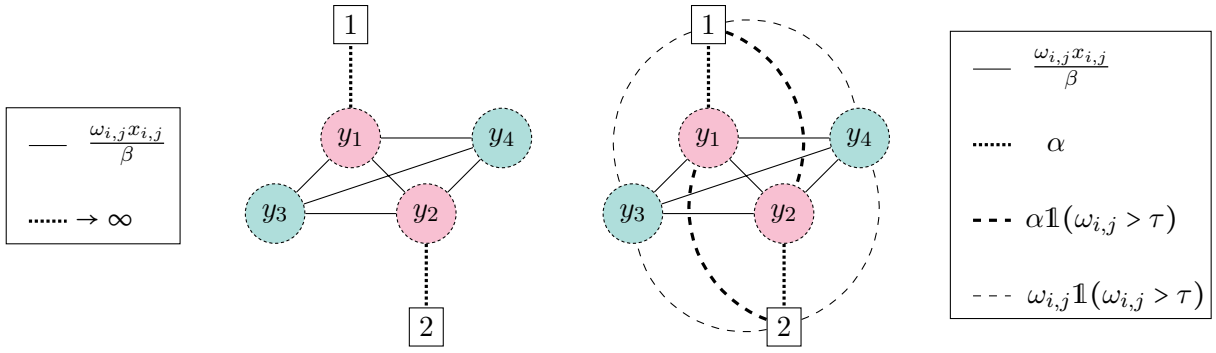
In details, we adopt a decoupling strategy allowing us to treat a multiple class problem as binary sub-problems. In addition, as we seek clusters attached to TFs, the label restriction to $\mathbb{T}$ is tackled by defining the set $\{s^{(1)}, \ldots, s^{(T)}\}$, with T binary vectors of length T. To emulate the second term in (6.25), their components are set to $s_t^{(t)} = 1$ and $s_j^{(t)} = 0$ if $j \neq t$. Let $\mathcal{Y} = \{y^{(1)}, \ldots, y^{(T)}\}$ be a set of T vectors. For all $t \in \mathbb{T}$, $y^{(t)} \in [0,1]^G$ contains the probabilities for nodes to be assigned to cluster t. Problem (6.25) is thus re-expressed as:

$$\underset{\mathcal{Y}\in([0,1]^G)^T}{\text{minimize}} \quad \sum_{t=1}^{T}\left(\sum_{(i,j)\in\mathbb{V}^2} \frac{\omega_{i,j}x_{i,j}}{\beta}\left(y_i^{(t)} - y_j^{(t)}\right)^2 + \sum_{i\in\mathbb{V},\, j\in\mathbb{T}} \mu_{i,j}\left(y_i^{(t)} - s_j^{(t)}\right)^2\right). \tag{6.26}$$

Independently from the choice of $\mu_{i,j}$ i.e. *hard*- or *soft*-clustering, the optimization of Problem (6.26) is illustrated with the graph structure of Figure 6.10. As displayed by Figure 6.10(b), the presence of strongly weighted edges between two TFs favors their merging. Merging is also possible for $\overline{\text{TF}}$ genes that also exhibit a strong weight with a TF. This copes with the fact that not all TFs are known in real biological datasets. Formulation (6.26) is an instance of the combinatorial Dirichlet problem and amounts to solving $T - 1$ systems of linear equations admitting a unique solution (Grady, 2006). The maximum probability arising from sub-problem t, $t \in \mathbb{T}$, defines each node label. The optimal cluster labeling $y^* = (y_i^*)_{1\leq i\leq G}$ is thus given by

$$\forall i \in \mathbb{V}, \quad y_i^* = \arg\max_{t\in\mathbb{T}} \; y_i^{(t)}. \tag{6.27}$$

As illustrated in Figure 6.11, computing optimal node clustering involving more than two classes (T in our case) can be decomposed into T-sub-problems. A given sub-problem t evaluates



(a) *hard*-clustering. (b) *soft*-clustering.

Figure 6.10 ~ GRAPH CONSTRUCTION FOR *hard* AND *soft*-CLUSTERING ~
Markers, TFs *and* $\overline{\text{TF}}$s *are square, pink and green nodes, respectively. In the* hard*-clustering (a),* $\mu_{i,j}$ *weights are set as in (6.21). The optimization constrains each* TF *to be assigned to the label of its native marker. In the* soft*-clustering (b), thanks to weights* $\mu_{i,j}$ *defined as (6.22), two clusters are merged if their respective* TFs *have strong weights, resulting in the same node cluster label. In the legend-box of the* soft*-clustering (b),* α *parameter refers to (6.23).*

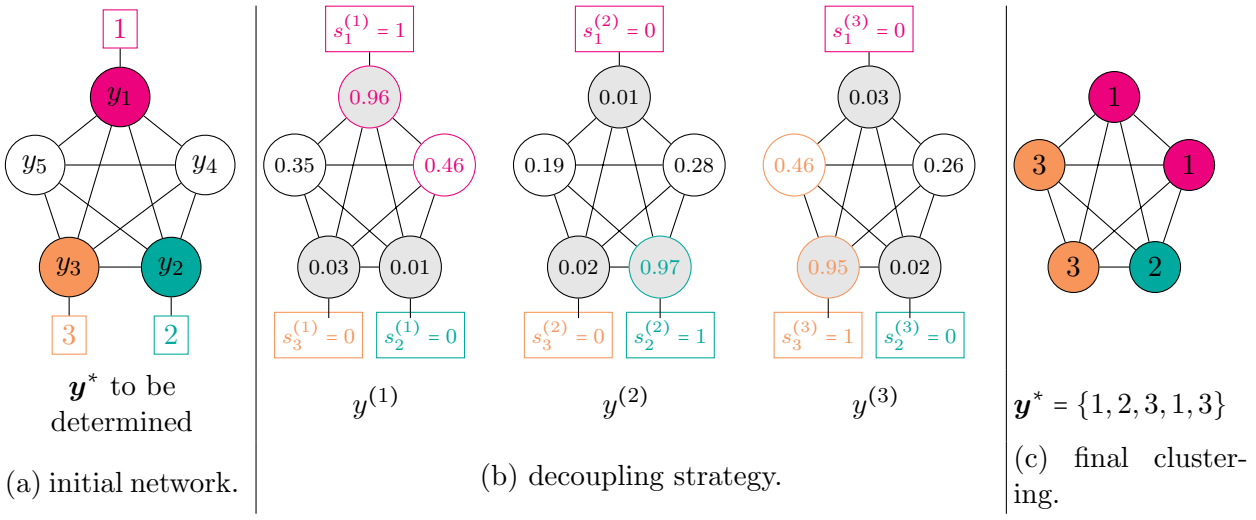
$y^{(t)}$ with respect to vector $s^{(t)}$. Its graph interpretation resorts to fixing marker label t to 1 and the others to 0, as described in Figure 6.11(b). Probability $y_i^{(t)}$ reflects the chance to reach the marker labeled by 1 first, for a random walker leaving node i in the graph. Higher weights encode preferable paths for the walker, and therefore drive the computed probabilities.

An approximate solution to Problem (6.20) yields the GRN after few iterations of alternating optimization between (6.24) and (6.26) — less than 20 with our datasets. Note that *BRANE Clust* with *hard*-clustering setting for $\mu_{i,j}$ converges in two iterations only, thus justifying the one-shot procedure (Algorithm 2) proposed in Section 6.2.2. As a result, our *BRANE Clust* algorithm with *soft*-clustering can be summed up as follows:

Algorithm 3: *BRANE Clust* with *soft*-clustering

Fix $\beta > 1$, $\tau \in [0,1]$ and $\lambda \in [0,1]$;
Initialize $x_0 = \mathbf{1}_G$;
for $k = 1, 2, \ldots$ **do**
  Compute the cluster node labeling $\boldsymbol{y}$ at iteration k using (6.26) and (6.27) ;
  Based on y, compute the edge labeling x at iteration k thanks to (6.4).

What is the computational complexity of BRANE Clust? Even for large-sized networks, *BRANE Clust* running times remain negligible with respect to weights computation. Networks of size 100 are obtained in few milliseconds while networks composed of 1000 to 5000 nodes are inferred in 1 s to 15 s. Running times are obtained using an Intel i7-3740QM @ 2.70GHz / 8 Gb RAM and Matlab 2011b. The costlier step is the *random walker* computation. Since the linear system is sparse, implementations with conjugate gradient drastically reduce the complexity, of at most



$\boldsymbol{y}^*$ to be determined

(a) initial network.

$y^{(1)}$ $y^{(2)}$ $y^{(3)}$

(b) decoupling strategy.

$\boldsymbol{y}^* = \{1, 2, 3, 1, 3\}$

(c) final clustering.

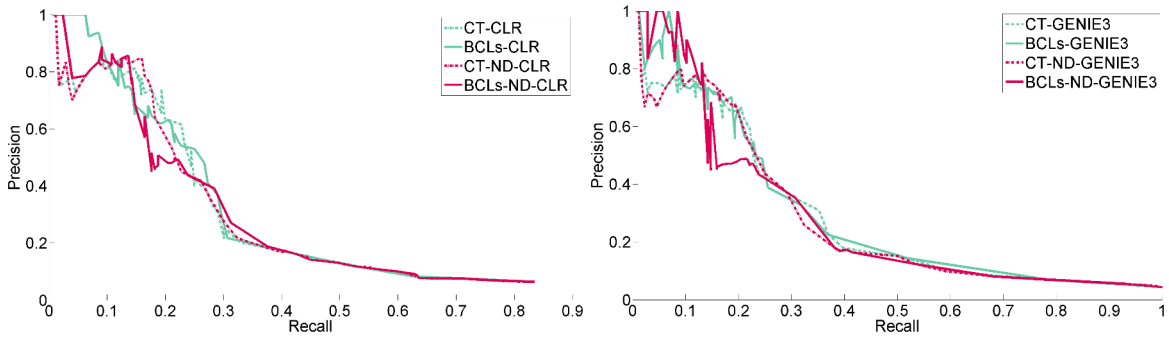
Figure 6.11 ~ GRAPH INTERPRETATION FOR *BRANE Clust* GENERALIZATION ~
To simplify, the principle is presented for the hard*-clustering but the principle is similar for* soft*-clustering. Markers,* TFs *and* $\overline{\text{TF}}$*s are square, filled and white nodes, respectively. Gene node to gene node edges are weighted by* $\frac{\omega_{i,j}x_{i,j}}{\beta}$ *while gene node to markers are weighted by* $\mu_{i,j}$*. The* T*-label problem is decomposed into* T *binary sub-problems by setting the component* t *of marker labels* $s^{(t)}, t \in \mathbb{T}$*, to one and the others to zero. Each sub-problem* t *leads to a probability for each node. The final node clustering corresponds to the label whose probability amidst the* T *sub-problems is maximal.*

$\mathcal{O}(G^3)$, where G is the number of nodes.

6.3.3 Objective results and biological interpretation

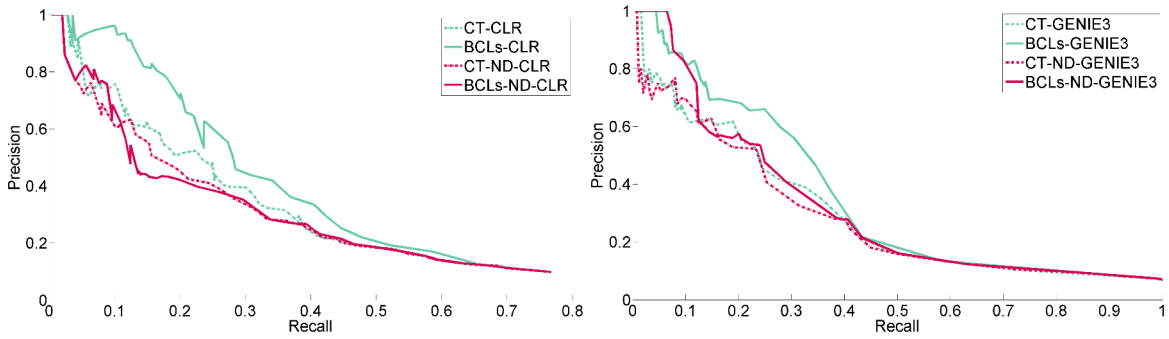
In this section, we present assessment of *BRANE Clust* with *soft*-clustering performed on simulated data (from DREAM4 and DREAM5 challenges) as well as on real *Escherichia coli* data from the DREAM5 challenge. Biological relevance of an inferred network by *BRANE Clust* is also evaluated, thus revealing the added value generated by our approach.

~ Numerical results on simulated data ~ As usual in this thesis, from a given set of initial weights, *BRANE Clust* with *soft*-clustering performance is compared to those obtain with the classical thresholding (CT). Initial weights are computed from simulated data, provided in both DREAM4 (datasets 1 to 5) and DREAM5 (dataset 1) challenges, using CLR or GENIE3. Post-processed CLR and GENIE3 weights by Network Deconvolution (ND) are also used in this evaluation. Each simulation yields a Precision-Recall (PR) curve, obtained by linearly varying the threshold parameter λ. Resulting PR curves are displayed in Figures 6.12 to 6.16. Numerical performance in terms of AUPRs and their relative gains are provided in Table 6.4.



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.12 ~ PR CURVES FOR THE DATASET 1 OF DREAM4 (*BRANE Clust–soft*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* soft*-clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*



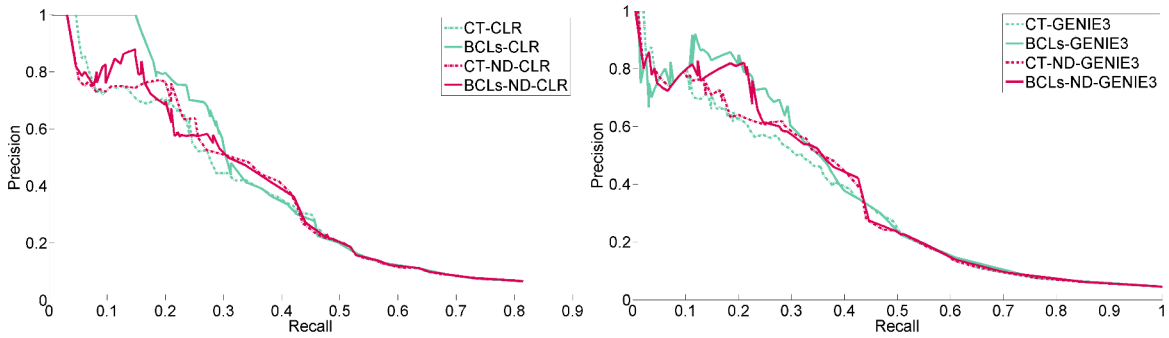
(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.13 ~ PR CURVES FOR THE DATASET 2 OF DREAM4 (*BRANE Clust–soft*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* soft*-clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

We observe that our proposed approach *BRANE Clust* with *soft*-clustering outperforms classical thresholding (CT) on all tested datasets and weights. Indeed, as highlighted in italic in Table 6.4(a), the two best AUPRs on each dataset are obtained using *BRANE Clust*. From a more global viewpoint on all datasets, average gains over CT reach 12.2 %, 12.8 %, 2.5 % and 8.1 % using CLR, GENIE3, ND-CLR and ND-GENIE3 weights, respectively (Table 6.4(b)).

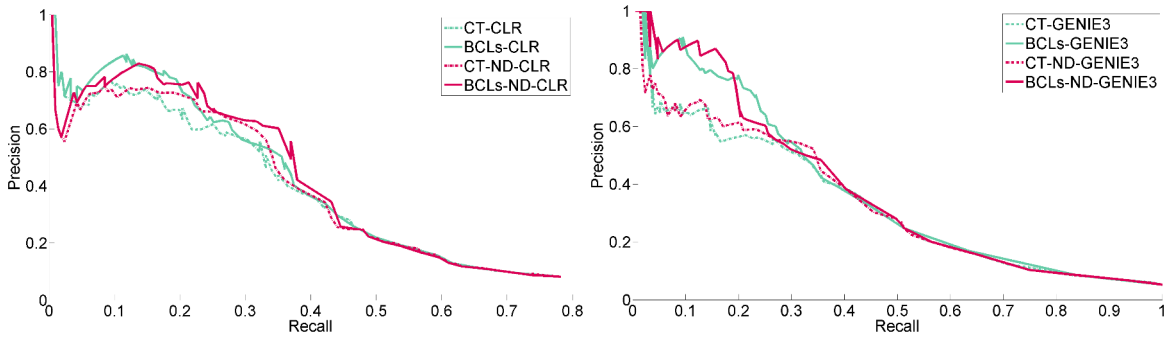
In addition, except for some cases, we can remark a significant improvement in the top-left part of the PR curves, for which networks contain less than 1000 edges. This observation is highlighted in the F-plots, which exhibit, for both CT and *BRANE Clust*, F-scores according to the number of edges in the network. F-score is an accuracy measure computed from Precision and Recall as:

$$F = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{6.28}$$



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.14 ~ PR CURVES FOR THE DATASET 3 OF DREAM4 (*BRANE Clust–soft*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* soft-*clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

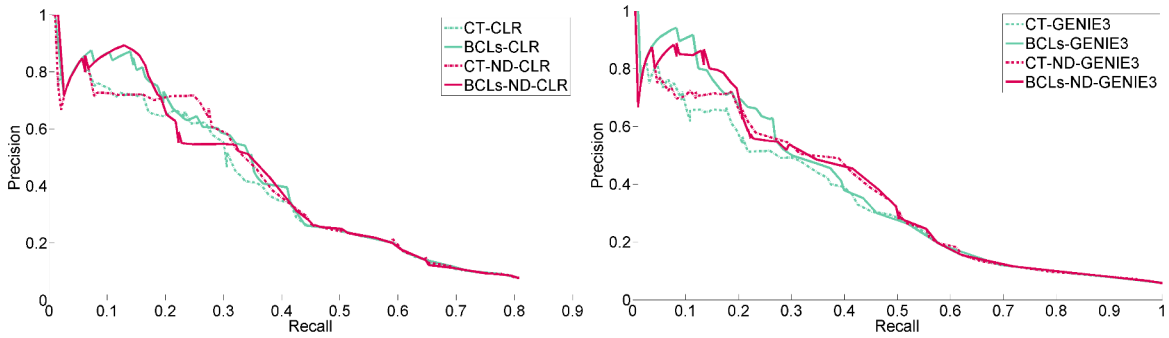


(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.15 ~ PR CURVES FOR THE DATASET 4 OF DREAM4 (*BRANE Clust–soft*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* soft-*clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

and represents the harmonic mean of Precision and Recall. We voluntarily restrict the construction of the curve to networks having from 10 to 1000 edges, as these networks are generally located in the top-left part of the PR curves, and are more interesting for gene interaction discovery. A typical example of such curves from dataset 2 is displayed in Figure 6.17 and all of them are provided at the end of this chapter (Figures 6.31 to 6.35 - p. 165 - p. 167). We observe — in the large majority of cases — higher F-scores for *BRANE Clust* compare to CT. These observations thus corroborate the fact that — in addition to favorable global performance — *BRANE Clust* especially refines classical thresholding results of networks expected as biologically relevant.

From a complementary perspective, comparisons of the post-processing itself (ND vs *BRANE Clust*) are in favor of our approach. Such a conclusion is drawn after comparing AUPRs obtained by *BRANE Clust* either on CLR or GENIE3 to CT on ND-CLR or ND-GENIE3, respectively.



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.16 ~ PR CURVES FOR THE DATASET 5 OF DREAM4 (*BRANE Clust–soft*) ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* soft*-clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

Dataset	1	2	3	4	5	Average
CT-CLR	0.256	0.275	0.314	0.313	0.313	0.294
BCLs-CLR	*0.275*	*0.337*	*0.360*	0.335	0.342	0.330
CT-GENIE3	0.269	0.288	0.331	0.323	0.329	0.308
BCLs-GENIE3	*0.287*	*0.348*	*0.364*	*0.371*	*0.367*	*0.347*
CT-ND-CLR	0.254	0.250	0.324	0.318	0.331	0.295
BCLs-ND-CLR	0.258	0.251	0.327	0.337	0.342	0.303
CT-ND-GENIE3	0.263	0.275	0.336	0.328	0.354	0.309
BCLs-ND-GENIE3	0.273	0.311	0.354	*0.373*	*0.370*	*0.336*

(a) AUPRs.

Dataset	1	2	3	4	5	Average
BCLs-CLR *vs* CT-CLR	7.4 %	22.5 %	14.6 %	7.0 %	9.3 %	12.2 %
BCLs-GENIE3 *vs* CT-GENIE3	6.7 %	20.8 %	10.0 %	14.9 %	11.5 %	12.8 %
BCLs-ND-CLR *vs* CT-ND-CLR	1.6 %	0.4 %	0.9 %	6.0 %	3.5 %	2.5 %
BCLs-ND-GENIE3 *vs* CT-ND-GENIE3	3.8 %	13.1 %	5.3 %	13.7 %	4.5 %	8.1 %

(b) Relative gains.

Table 6.4 ~ NUMERICAL PERFORMANCE ON DREAM4 (*BRANE Clust–soft*) ~ *(a) Area Under PR curve (AUPR) obtained using CT or BRANE Clust with* soft*-clustering (BCLs) on CLR, ND-CLR, GENIE3 and ND-GENIE3 weights. Weights are computed for each dataset (1 to 5) of the DREAM4 multifactorial challenge. Average AUPR are also reported as well as the two maximal improvements (in italics). (b) Relative gains obtained by comparing BRANE Clust with* soft*-clustering to CT.*

Dataset	1	2	3	4	5	Average
BCLs-CLR *vs* CT-ND-CLR	8.3 %	34.8 %	11.1 %	5.3 %	3.3 %	12.6 %
BCLs-GENIE3 *vs* CT-ND-GENIE3	9.1 %	22.5 %	7.1 %	13.1 %	−3.4 %	9.7 %

Table 6.5 ~ Post-processing performance on DREAM4 (*BRANE Clust–soft*) **~**
Relative gains computed using AUPRs provided in Table 6.4(a) are given for BRANE Clust with soft*-clustering using CLR (resp. GENIE3) weights compared to CT using ND-CLR (resp. ND-GENIE3).*

Indeed, relative gains, summarized in Table 6.5, reach in average over the five datasets, a percentage of 12.6 and 9.7 based on CLR and GENIE3 weights, respectively.

In view on the promising results obtained on the five simulated datasets of the DREAM4 challenge, *BRANE Clust* is then assessed in a more practical context — always in a step-by-step strategy — firstly using the realistic simulated dataset of DREAM5.

As previously, CT and *BRANE Clust* are compared in terms of AUPR, computed from PR curves obtained using CLR, GENIE3, ND-CLR or ND-GENIE3 weights. PR curves are displayed in Figure 6.18 and corresponding AUPRs and relative gains are reported in Table 6.7.

	AUPR	Gain		AUPR	Gain
CT-CLR	0.252	19.4 %	CT-ND-CLR	0.272	6.2 %
BCLs-CLR	0.301		BCLs-ND-CLR	0.289	
CT-GENIE3	0.283	18.6 %	CT-ND-GENIE3	0.313	10.2 %
BCLs-GENIE3	0.336		BCLs-ND-GENIE3	0.345	

Table 6.6 ~ Numerical performance on the dataset 1 of DREAM5 (*BRANE Clust–soft*) **~**
Area Under Precision-Recall curve (AUPR) obtained using CT or BRANE Clust with soft*-clustering on CLR, ND-CLR, GENIE3 or ND-GENIE3 weights computed from dataset 1 of the DREAM5 challenge. Relative gains between CT and BRANE Clust are also reported.*

BRANE Clust offers refined results compared to CT with a maximal improvement reaching 19.4 %. Improvement are particularly significant in the top-left part of PR curves, corresponding to networks with less than 1000 edges, in this dataset. This observation is sustained when F-plots (Figure 6.19) are considered. They represent F-measures, computed as in (6.28), according to the number of edges in the network — restricted to a range from 10 to 1000 edges. This choice was driven by the fact that we are focused on biologically interpretable and relevant networks, expected with less than 1000 edges. Curves in Figure 6.19 highlight higher F-scores for *BRANE Clust* than for CT when they are compared on small but biologically interpretable networks.

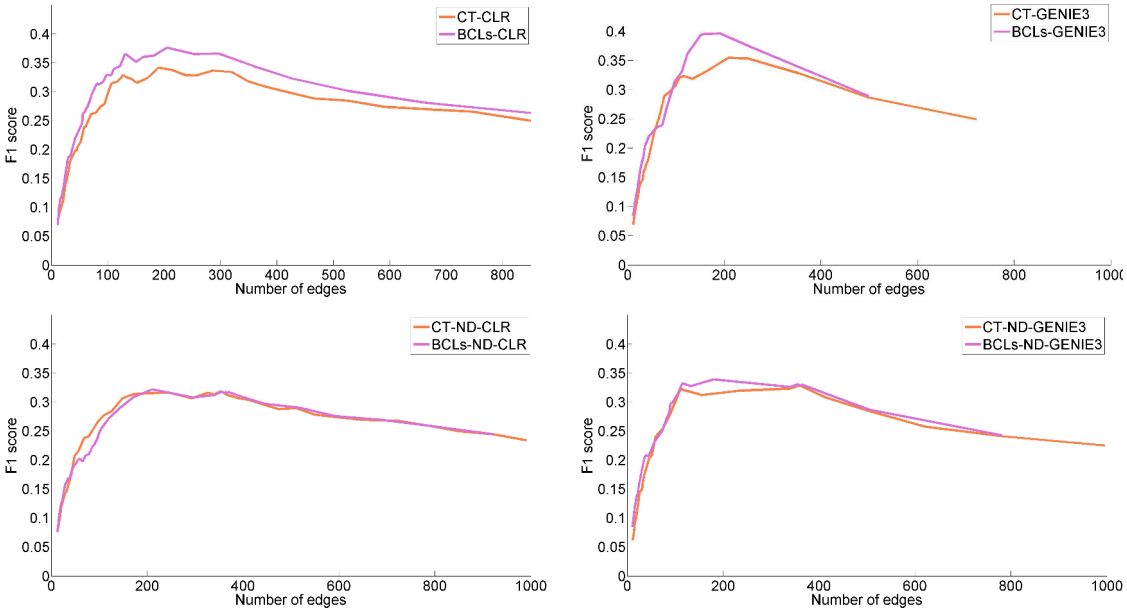
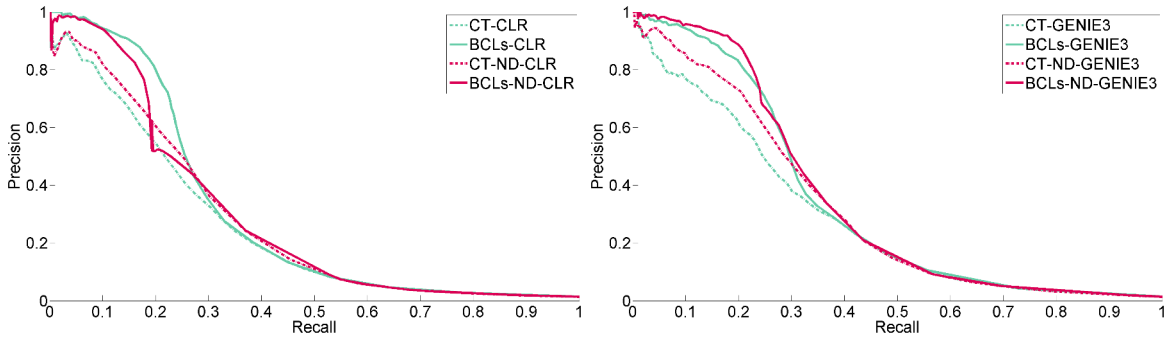


Figure 6.17 ~ *F*-PLOTS FOR THE DATASET 2 OF DREAM4 (*BRANE Clust-soft*) **~** *Curves depicting F-scores according to the number of edges (in a range from 10 to 1000), generated by CT or BRANE Clust on CLR, GENIE3, ND-CLR or ND-GENIE3 weights.*



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.18 ~ PR CURVES FOR THE DATASET 1 OF DREAM5 (*BRANE Clust–soft*) **~** *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* soft*-clustering on (a) CLR and ND-CLR weights or (b) GENIE3 and ND-GENIE3 weights.*

BRANE Clust being now validated on more or less realistic simulated data, the crucial transition to real data can be considered.

~ *Numerical results on real data* **~** For this purpose, we used the dataset 3 provided by the DREAM5 challenge. As detailed in Section 3.2.1, this dataset encompasses a compendium of real transcriptomic data coming from various studies on the bacteria *Escherichia coli*. From

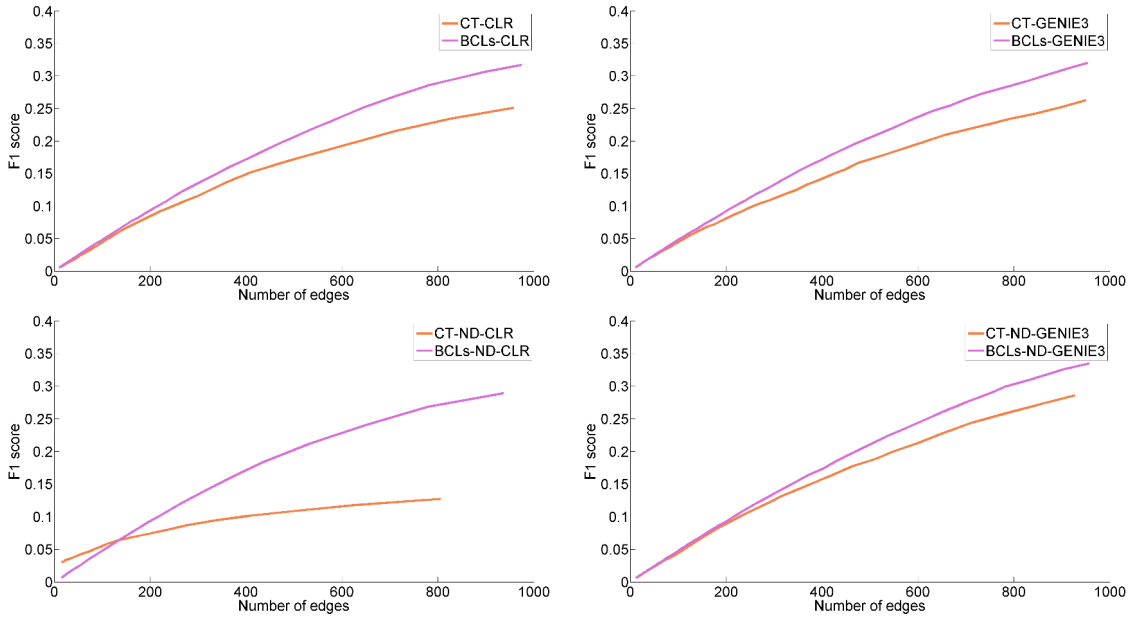


Figure 6.19 ~ *F*-PLOTS FOR THE DATASET 1 OF DREAM5 (*BRANE Clust-soft*) ~ *Curves depicting F-scores according to the number of edges (in a range from 10 to 1000), generated by CT or BRANE Clust on CLR, GENIE3, ND-CLR or ND-GENIE3 weights.*

this dataset, a complete weighted graph is generated thanks to either CLR or GENIE3. By varying the λ parameter, we then evaluate networks generated by CT or *BRANE Clust* through the obtained PR curves and their respective AUPR. Resulting PR curves are displayed in Figure 6.20 while Table 6.7 summarizes numerical performance in terms of AUPR and relative gain.

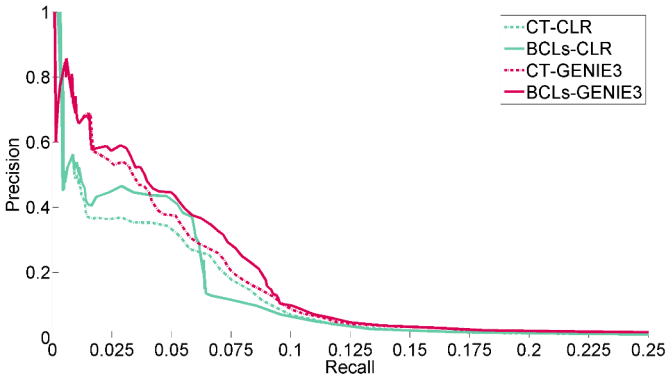


Figure 6.20 ~ PR CURVES FROM *Escherichia coli* DATASET ~ *Precision-Recall (PR) curves obtained using CT or BRANE Clust with* soft-*clustering on CLR or GENIE3 weights.*

Results obtained from real *Escherichia coli* experiments exhibit a global improvement reaching gains about 6 % and 10 % using CLR and GENIE3 initial weights, respectively. Unlike previous results, improvements are not focused on the top-left part of the PR-curves, but for lower Precision. This unexpected results can be discussed. Indeed, in this dataset, the top-left

	AUPR	Gain		AUPR	Gain
CT-CLR	0.0378	5.5 %	CT-GENIE3	0.0488	9.8 %
BCLs-CLR	0.0399		BCLs-GENIE3	0.0536	

Table 6.7 ~ NUMERICAL PERFORMANCE OF *BRANE Clust* ON THE *Escherichia coli* DATASET ~ *Area Under Precision-Recall curve (AUPR) obtained using CT or BRANE Clust with soft-clustering on CLR or GENIE3 weights computed from dataset 3 of the DREAM5 challenge. Relative gains between CT and BRANE Clust are also reported.*

part of the PR curves (Recall from 0 to 0.01) corresponds to very small graphs with less than 50 edges. Although such graphs are reliable, they are poorly informative and thus rarely expected by biologists, because they generally correspond to known results. We observe that interesting graphs, containing from 50 to 1000 edges, are located in a Recall range of $[0.01, 0.08]$. However, for this given range of Recall, corresponding Precision values drop drastically below 0.5. This trade-off between network size and reliability is thus problematic and has to be preferentially resolved. Notably, *BRANE Clust* offers significant improvement in this area of higher importance, in which networks of required size become more reliable as Precision increases. F-plots displayed in Figure 6.21 argue for this observation. These results are thus in favor of our proposed approach *BRANE Clust* and show an interest for combining clustering and inference. In addition, while comparisons between *hard*-clustering and *soft*-clustering versions of *BRANE Clust* can be sometimes debated, *soft*-clustering version of *BRANE Clust* generally provides better numerical results than the *hard*-clustering version.

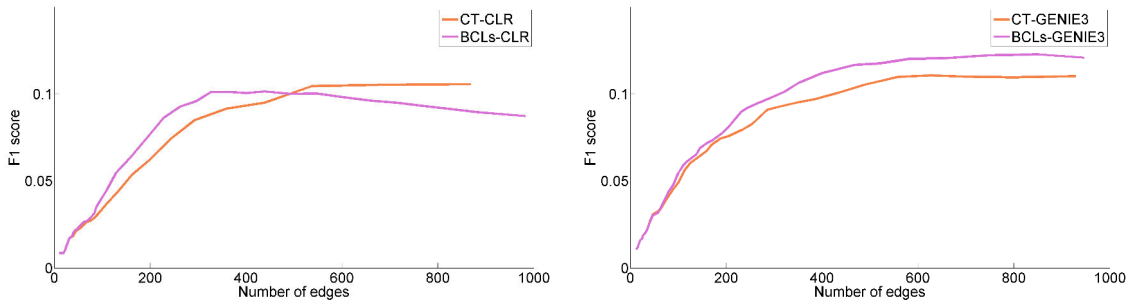


Figure 6.21 ~ F-PLOTS FOR THE DATASET 3 OF DREAM5 (*BRANE Clust-soft*) ~ *Curves depicting F-scores according to the number of edges (in a range from 10 to 1000), generated by CT or BRANE Clust on CLR or GENIE3 weights.*

Although numerical results are promising on real data, an additional validation — from a biological viewpoint — is required. Indeed, the tininess of obtained Precision and Recall for networks of interest persuade us to perform additional validation and assessment to vouch for good performance of *BRANE Clust*, more rigorously. We thus dedicated the next section to the biological evaluation of *BRANE Clust*.

~ *Biological validation* ~ We evaluate the biological interest of *BRANE Clust* by comparing inferred networks of *Escherichia coli* using CT or *BRANE Clust* on GENIE3 weights. For this purpose, we select CT and *BRANE Clust* networks composed of 236 edges which provides the best compromise in size and improvement. As summarized in Figure 6.22, we firstly compare network characteristics, in terms of Precision, Recall, number of TP and FP edges in common or specific to CT and *BRANE Clust*.

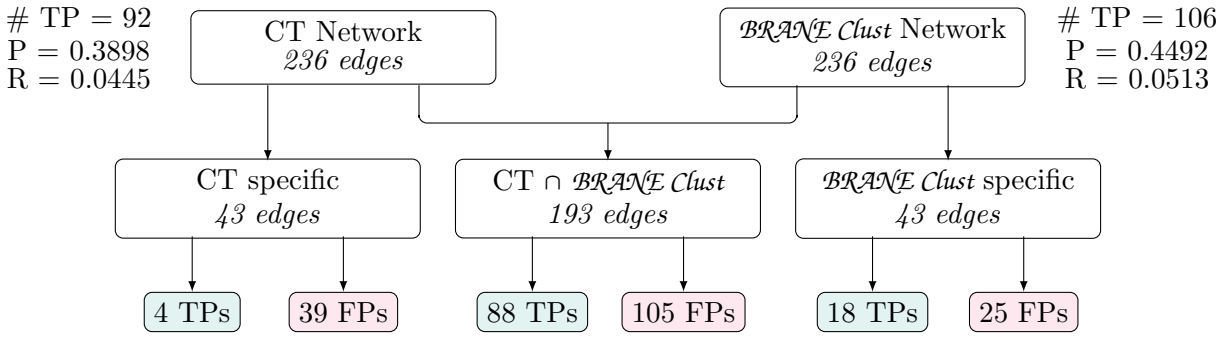


Figure 6.22 **~** CT AND *BRANE Clust Escherichia coli* NETWORK CHARACTERISTICS **~** *Networks are generated with CT or BRANE Clust on pre-computed GENIE3 weights from the* E. coli *dataset.*

Network generated by *BRANE Clust* is displayed, at the end of this chapter, in Figure 6.30 - p. 164. True and false edges are distinguishable by their color, respectively in pink and green. In addition, solid edges refer to commonly inferred edges by both CT and *BRANE Clust*, while dashed edges encodes those specifically selected by *BRANE Clust*.

Comparing equal-size networks, we observe that *BRANE Clust* generates a more reliable network with a Precision of about 45 % against 39 %. Putting the 193 common edges aside, 43 edges are thus specifically inferred by CT or *BRANE Clust*— namely, about 20 % of the network. Among the 43 edges specifically inferred by CT, only four are also recovered in the ground truth. *BRANE Clust* makes the difference: among its 43 specific edges, about 42 % are true, being 18 TP edges. Based on this comparison, *BRANE Clust* seems to generate more reliable networks. However, network reliability is not the unique criterion to be assessed. Indeed, predictive power should also be taken into account. For this purpose, it is interesting to evaluate the biological relevance of potential wrongly inferred edges (or predictions), — 25 with *BRANE Clust* and 39 with CT.

As mentioned in Section 3.2.2, prediction analyses are performed thanks to various databases such as RegulonDB (Gama-Castro *et al.*, 2016), EcoCyc (Keseler *et al.*, 2013) or STRING (Franceschini *et al.*, 2013). Note that, as two TF-TF symmetric relationships are found, the study is carried out on 23 predictions. Among them — *rhaT-rahR*, *gadE-yccB*, *deoR-ybjG*, *melR-yghZ*, *mprA-ygaZ*, *cbl-cysI*, *cbl-cysA*, *cbl-cysM*, *cbl-cysD*, *cbl-yciW*, *lrp-aroG*, *lrp-argA*, *lrp-yliJ*, *lrp-trpL*, *lrp-ilvC*, *lrp-nadA*, *allS-gcl*, *mhpR-glcC*, *nac-sdaC*, *nac-rutA*, *zraA-ilvY*, *fis-*

rpsF, *galS-mglA* — 6 are recovered as direct links in the STRING database for which details are reported in Table 6.8.

Prediction	Co-O	Co-E	Co-M	N	CS
mprA-ygaZ	-	-	-	0.606	0.606
cbl-yciW	0.211	0.364	0.321	-	0.629
deoR-ybjG	-	0.149	-	0.671	0.708
mhpR-glcC	-	0.670	0.116	-	0.745
galS-mglA	-	0.915	0.403	0.370	0.975
rhaT-rhaR	0.699	0.678	0.867	-	0.985

Table 6.8 ~ SIGNIFICANT STRING SCORES FOR *BRANE Cut* PREDICTIONS ~
STRING scores evaluate functional links between two genes and involve here probabilities based on co-occurrence across genomes (Co-O), co-expression (Co-E), co-mentioned in PubMed abstracts (Co-M), neighborhood in the genome (N). Combined Score (CS) is the final score taking account all the probabilities.

Among the 17 remaining predictions, two aspects can be considered: isolated or grouped links. The first category encompasses one-TF-one-target links *gadE-yccB*, *melR-yghZ*, *allS-gcl*, *zraR-ilvY* and *fis-rpsR*. Including indirect effects, three among them make sense at a larger scale of the regulation. Firstly, the relationship between *allS* and *gcl* — in addition to their proximity in the genome of *E. coli* — results in the action of *allR* on these two genes. Similarly, the TF *crp*, regulating the transcription of several catabolite-sensitive operons, both regulate *zraR* and *ilvY*. Finally, although this link has not been identified, the *fis*-rpsF link is not nonsensical. Indeed, *fis* is known to regulate many genes involved in large mechanisms such as the organization and the maintenance of nucleotide structure. The gene *rpsF* takes part in these mechanisms and two similar genes, *rpsO* and *rpsI*, have been identified as targets of *fis*. The second category of predictions, characterized by a one-TF-multiple-targets scheme, makes more sense in terms of co-expressed genes. Notably, predicted targets for the TF *cbl* are *cysA*, *cysd*, *cysI* and *cysM*, which are known to be co-expressed genes. Similarly, genes *aroG* and *nadA* seem to be co-expressed and are both linked to *lrp*. The latter is also linked to *ilvC*, which is — in the *E. coli* genome — close to *ilv* operons, themselves regulated by *lrp*. As a result, even if all regulatory links are not validated as such, about half on the 25 predictions make sense and seem to be biologically relevant. Hence, they become plausibly good candidates for biological experiments. Figure 6.23 summarizes the biological assessment of the *BRANE Clust* predictions.

What can we say about clustering results? *BRANE Clust* returns at the same time a GNR and a gene clustering. We thus compare clustering results obtained from the *Escherichia coli* dataset, at the same time as the generated network displayed in Figure 6.30. For this purpose, *BRANE Clust* clustering is compared with WGCNA (Langfelder and Horvath, 2008) and X-means clustering (Pelleg and Moore, 2000). The latter, an extension to K-means (Steinhaus, 1956; MacQueen, 1967) with an optimal number of classes, not specific to biological applications, was used recently (Wang *et al.*, 2012a; Halleran *et al.*, 2015) in this context. Partitions are graded pair-wise, using

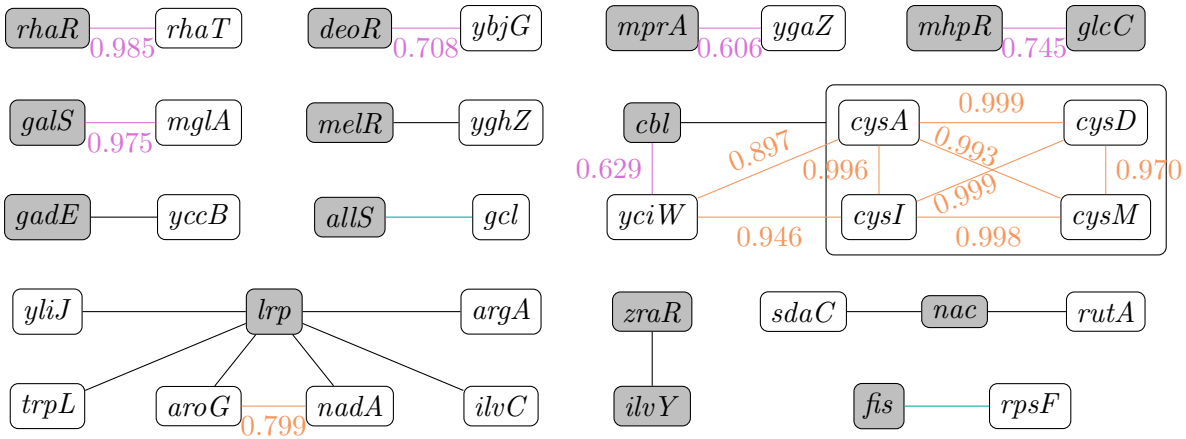


Figure 6.23 ~ *BRANE Clust* PREDICTIONS AND STRING VALIDATION ~
All links specifically inferred by BRANE Clust are reported as well as significant CS scores obtained with STRING. Purple scores and edges refer to direct links found in STRING database while orange scores and edges refer to direct links between targets. Green edges refers to (yet) unidentified predictions for which the exploration of databases reveals a plausible biological relevance.

the Variation of Information (VI, Meilă (2007)), a metric closely related to mutual information (detailed in Section 3.2.3). *BRANE Clust* modules (genes arranged around TFs) differ from those in WGCNA or X-means. WGCNA provides 18 modules, X-means 17 clusters, and 322 for *BRANE Clust* partitioning. Hence, we expect a poor pairwise overlap between these methods, as confirmed in Figure 6.24 with significantly non-null VI measures.

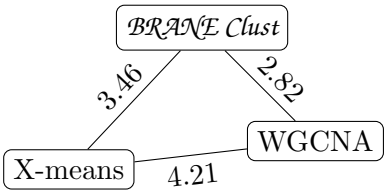


Figure 6.24 ~ INTRINSIC CLUSTERING EVALUATION OF *BRANE Clust* ~
Pairwise VI (Variation of Information) measures for BRANE Clust, WGCNA and X-means.

However, with a closer number of clusters, WGCNA and X-means surprisingly exhibit the largest VI (4.21), thus the least similarity. The best partition overlap (2.82) is observed between WGCNA and *BRANE Clust*, despite the gap in cluster amount. An external validation with biologically-sound groups of genes from a validated database may be more pertinent. It is built from operons — we recall that operons denote transcriptional units of genes controlled by a single promoter, akin to our TF-centric clusters — identified in RegulonDB (Gama-Castro *et al.*, 2016). All significant operons, containing at least 5 genes, compose the ground truth. It splits a subset of 803 genes into 123 groups. We compare this partitioning to those of *BRANE Clust*, WGCNA and X-means on the same gene subset in Table 6.9. A smaller VI (higher similarity) is found for *BRANE Clust*, suggesting that its partitioning is nigher in terms of operon structure.

	BRANE Clust	WGCNA	X-means
# of clusters	90	18	17
VI (vs RegulonDB)	1.05	1.10	1.14

Table 6.9 ~ External clustering/operon evaluation of *BRANE Clust* ~
VI (Variation of Information) measures for BRANE Clust, WGCNA and X-means vs RegulonDB.

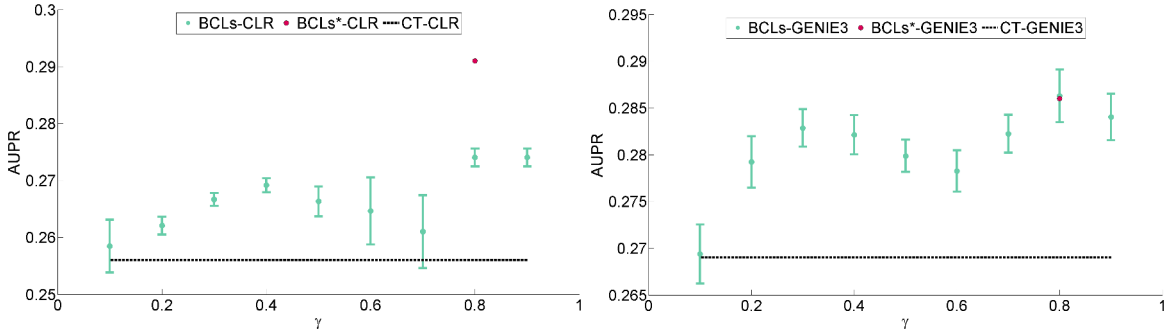
~ *Parameter settings* ~ Our *BRANE Clust* model (6.20) involves four parameters: λ, β, τ and α. We recall that the threshold parameter λ is common with the classical thresholding. It is used, in comparative studies, to construct PR curves. In practice, no automatic setting is known and users set it often manually in order to recover relatively small networks (less than 1000 edges) for which a biological interpretation is feasible. The three other parameters take part in the clustering *a priori*. Specifically, $\beta > 1$ controls the influence of the clustering in the inference. In all simulations, β was set to 2 and provides good compromise, whatever the dataset and the weights used. Thus, fixing $\beta = 2$ imparts a satisfying start point. Parameters $\tau \in [0, 1]$ and $\alpha > 0$ drive the clustering itself, notably regarding the cluster merging. Indeed, τ is a threshold parameter answering to the question: *Should these clusters merge?* If the answer is positive, α reflects the strength bestowed to the merge promotion. As mentioned in (6.23), the latter parameter can be set automatically. Note that, while it provides a correct start point for α parameter setting, results can be refined by adjusted the parameter according to the considered initial weights. For instance:

$$\alpha = \frac{\sum_{(i,j)\in\mathbb{T}^2} \mathbb{1}(\omega_{i,j} > \tau)}{\overline{\omega_{i,j}}}, \tag{6.29}$$

where $\overline{\omega_{i,j}}$ is the median of non-zero TF-TF weights. This setting was used for simulation on the Dataset 1 of DREAM5, for which the metric choice appears more sensitive. Let us now focus on the choice of τ, where values close to 0 or 1 would disfavor either clustering or inference, unbalancing the performance of *BRANE Clust*. Hence, a suitable range for τ resides around the central inter-quartile range. In our simulations, τ was set to 0.3 and 0.8 in simulated and real datasets, respectively. The motivation follows: DREAM4 *in silico* data is generated with GeneNetWeaver (Schaffter *et al.*, 2011) and is based on true networks. A perfect knowledge of TFs is thus available and simulated gene expressions are considered more reliable. Hence, we have more confidence in strong edge weights for the cluster fusion task. With real data, conversely, uncertainty in experimental gene expressions and partial knowledge of TFs are an incentive for lower levels. The latter tend to redeem lower weights, affected by experimental biases and variability.

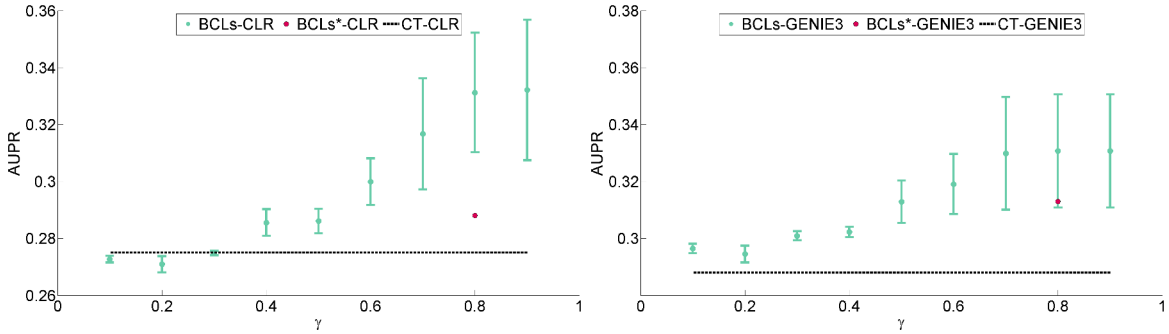
As a result, putting λ and α aside, only β and τ have to be fixed. It is thus judicious to perform a sensitivity analysis for both β and τ. For this purpose, a grid-search strategy was employed and performed on two kind of weights (CLR and GENIE3). The parameter τ varies between 0.1 and 0.9 with a 0.1 step. The β varies between 1.1 and 2 with a 0.1 step, and between 2 and 5 with a unit step. AUPRs for each couple of parameters are computed and results are compiled in Figures 6.25 to 6.29. For each τ, we report the average AUPR and its standard

deviation over β.



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

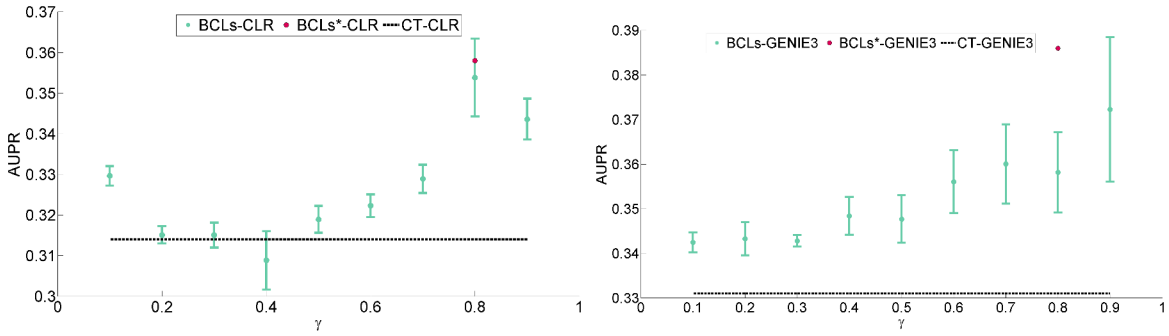
Figure 6.25 ~ SENSITIVITY ANALYSIS OF τ AND β ON THE DATASET 1 OF DREAM4 ~ *Assessment of parameter effects on AUPRs obtained using BRANE Clust on (a) CLR and (b) GENIE3 weights. For each τ, results obtained with BRANE Clust are given in terms of average AUPR and standard deviation over β. BCLs*- refers to the AUPR results obtained with BRANE Clust using the parameter setting described in this current section. AUPRs obtained with CT are also recalled.*



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

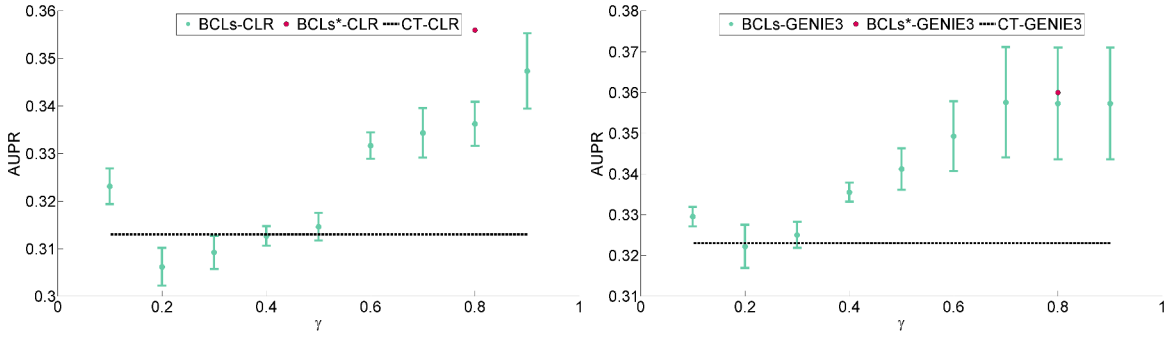
Figure 6.26 ~ SENSITIVITY ANALYSIS OF τ AND β ON THE DATASET 2 OF DREAM4 ~ *Assessment of parameter effects on AUPRs obtained using BRANE Clust on (a) CLR and (b) GENIE3 weights. For each τ, results obtained with BRANE Clust are given in terms of average AUPR and standard deviation over β. BCLs*- refers to the AUPR results obtained with BRANE Clust using the parameter setting described in this current section. AUPRs obtained with CT are also recalled.*

On the five datasets, we observe that, except for only few cases, the average AUPR obtained using *BRANE Clust* — at different τs — is significantly higher than the CT AUPR, when they are compared using either CLR or GENIE3 as initial weights. Although the variability over β often increases with τ, higher τ yield significantly better AUPRs. The increase in β variability



(a) Based on CLR weights. (b) Based on GENIE3 weights.

Figure 6.27 ~ SENSITIVITY ANALYSIS OF τ AND β ON THE DATASET 3 OF DREAM4 ~ *Assessment of parameter effects on AUPRs obtained using BRANE Clust on (a) CLR and (b) GENIE3 weights. For each τ, results obtained with BRANE Clust are given in terms of average AUPR and standard deviation over β. BCLs*- refers to the AUPR results obtained with BRANE Clust using the parameter setting described in this current section. AUPRs obtained with CT are also recalled.*

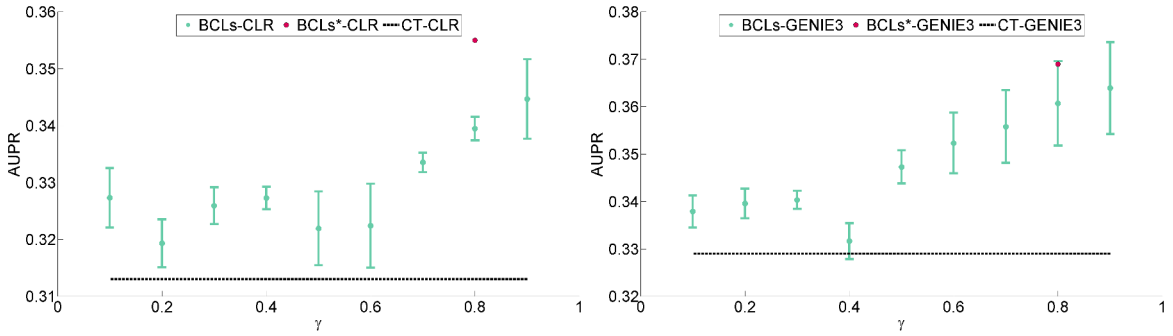


(a) Based on CLR weights. (b) Based on GENIE3 weights.

Figure 6.28 ~ SENSITIVITY ANALYSIS OF τ AND β ON THE DATASET 4 OF DREAM4 ~ *Assessment of parameter effects on AUPRs obtained using BRANE Clust on (a) CLR and (b) GENIE3 weights. For each τ, results obtained with BRANE Clust are given in terms of average AUPR and standard deviation over β. BCLs*- refers to the AUPR results obtained with BRANE Clust using the parameter setting described in this current section. AUPRs obtained with CT are also recalled.*

with τ may be explained by the selectivity of cluster merging. Low τ levels significantly trigger cluster fusion. The reduction in the number of labels diminishes the impact of β.

As demonstrated by our results, satisfactory trade-offs are obtained with this parameter setting for all experiments, whatever the data (size, weights, number of TFs) and the initial weights. Note that, presented results are not individually the best, and additional refinement



(a) Based on CLR weights. *(b) Based on GENIE3 weights.*

Figure 6.29 ~ SENSITIVITY ANALYSIS OF τ AND β ON THE DATASET 5 OF DREAM4 ~ *Assessment of parameter effects on AUPRs obtained using BRANE Clust on (a) CLR and (b) GENIE3 weights. For each τ, results obtained with BRANE Clust are given in terms of average AUPR and standard deviation over β. BCLs*- refers to the AUPR results obtained with BRANE Clust using the parameter setting described in this current section. AUPRs obtained with CT are also recalled.*

can be obtained by minor parameter adjustment. Notwithstanding, based on our simulations, we advise to fix 2 and 0.3 as efficient initial choice for β and τ, respectively.

6.4 Conclusions on *BRANE Clust*

BRANE Clust is our first step toward a better integrated framework for network analysis. Inference is coupled with clustering for an enhanced interpretation of inferred modules, more directly helping a biological functional investigation. Moreover, its main advantage over *BRANE Cut* resides in more intuitive and versatile cluster merging options, which we do not have fully explored yet. As *BRANE Cut* and *BRANE Relax*, it is a generic post-processing tool working on any complete weighted network. It favors edges both having higher weights and linking nodes belonging to a same cluster. The proposed cost function is solved through an alternating optimization procedure involving an explicit solution for the edge selection while the gene clustering is obtained *via* a random walker algorithm. Numerical performance on synthetic and real datasets (DREAM4 and DREAM5) shows significant improvement over state-of-the-art method. Biological relevance is also validated in depth on the *Escherichia coli* network.

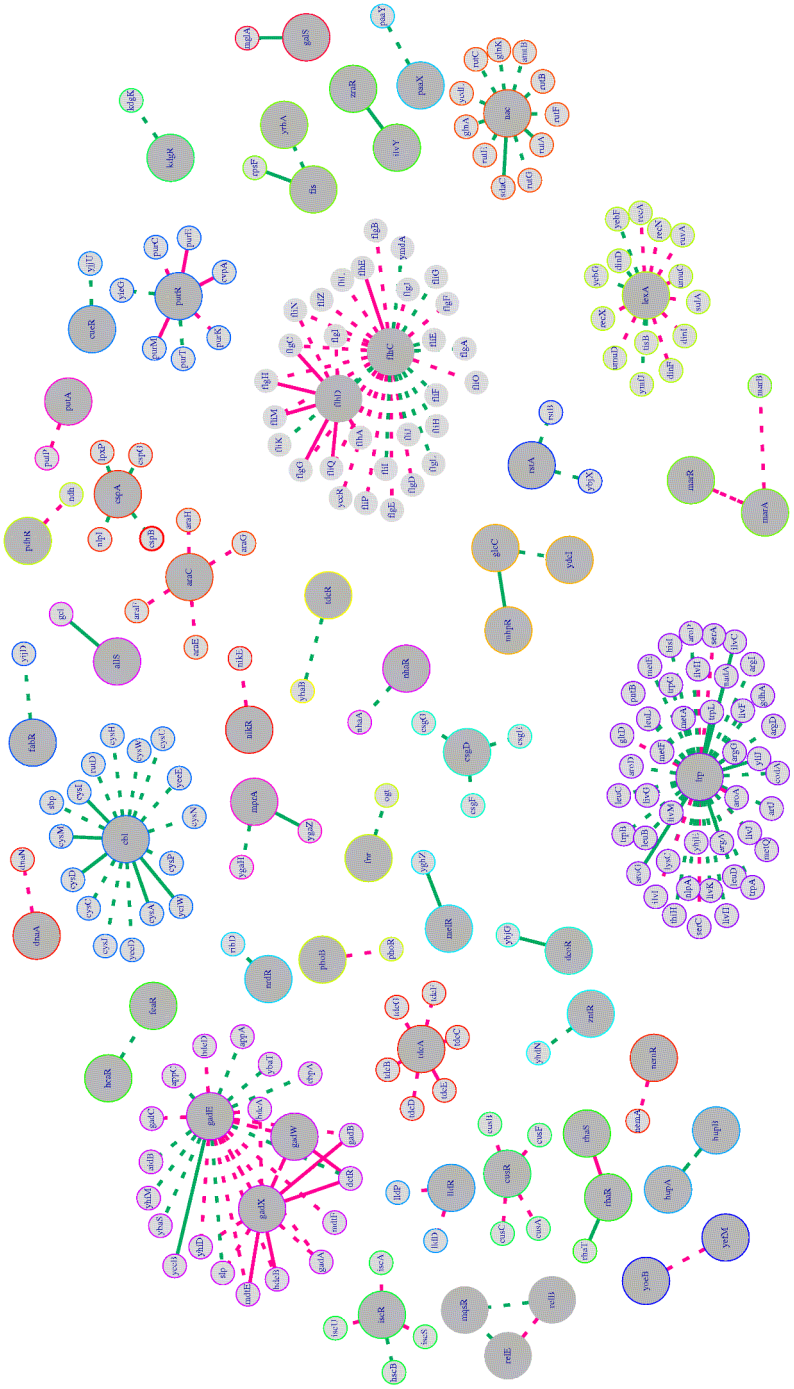


Figure 6.30 ~ INFERRED *Escherichia coli* NETWORK WITH *BRANE Clust* ~
Network built using BRANE Clust on GENIE3 weights and containing 236 edges. Large dark gray nodes refers to TFs. *Inferred edges also reported in the ground truth are colored in pink while predictive edges are green. Dashed edges correspond to a link inferred by both BRANE Clust and CT while solid links refer to edges specifically inferred by BRANE Clust. Colored node contour refers to cluster affiliations.*

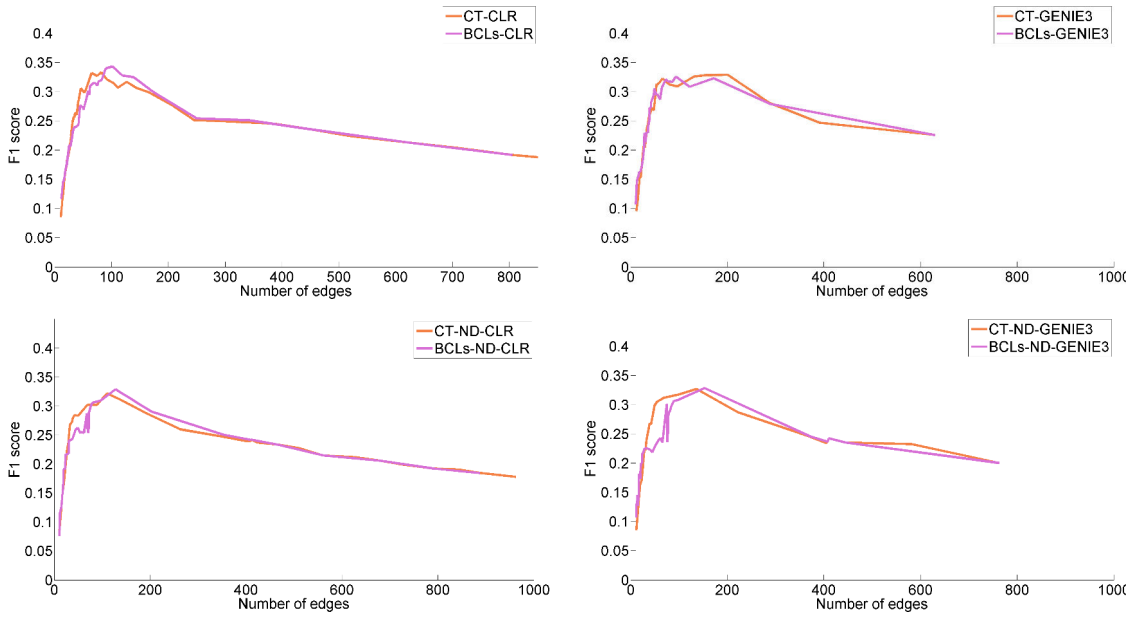


Figure 6.31 ~ *F*-PLOTS FOR THE DATASET 1 OF DREAM4 (*BRANE Clust-soft*) ~ *Curves depicting F-scores according to the number of edges (in a range from 10 to 1000), generated by CT or BRANE Clust on CLR, GENIE3, ND-CLR or ND-GENIE3 weights.*

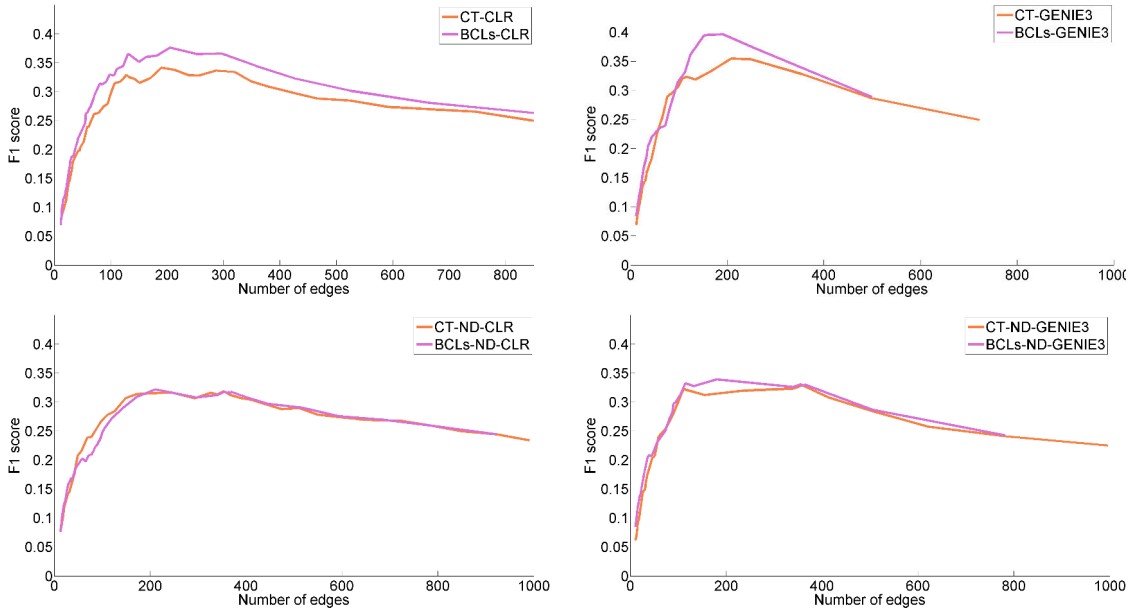


Figure 6.32 ~ *F*-PLOTS FOR THE DATASET 2 OF DREAM4 (*BRANE Clust-soft*) **~** *Curves depicting F-scores according to the number of edges (in a range from 10 to 1000), generated by CT or BRANE Clust on CLR, GENIE3, ND-CLR or ND-GENIE3 weights.*

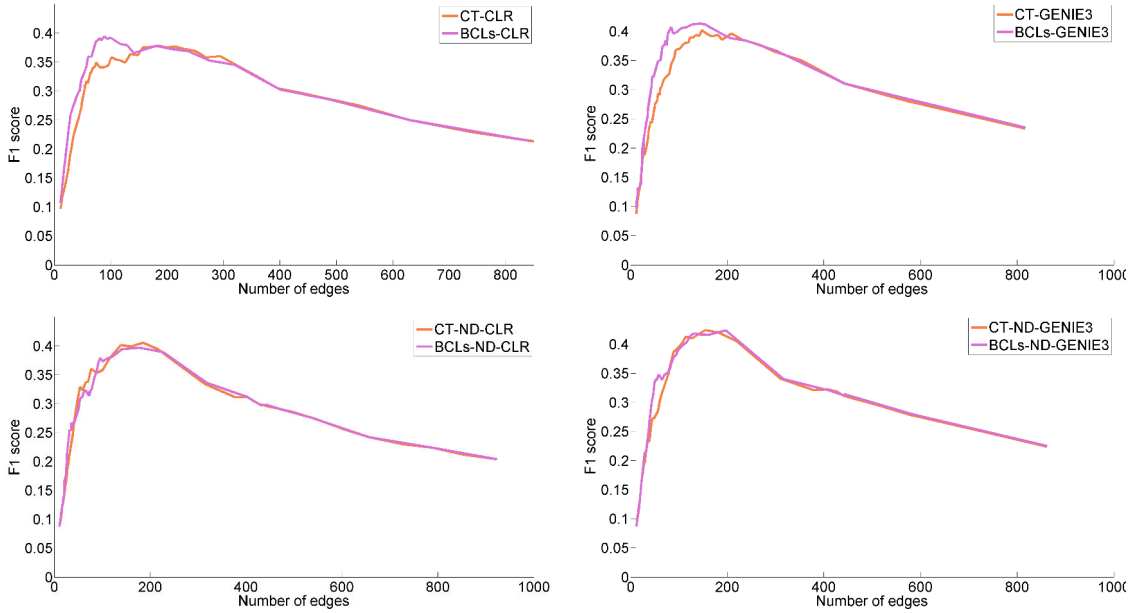


Figure 6.33 ~ *F*-PLOTS FOR THE DATASET 3 OF DREAM4 (*BRANE Clust-soft*) **~** *Curves depicting F-scores according to the number of edges (in a range from 10 to 1000), generated by CT or BRANE Clust on CLR, GENIE3, ND-CLR or ND-GENIE3 weights.*

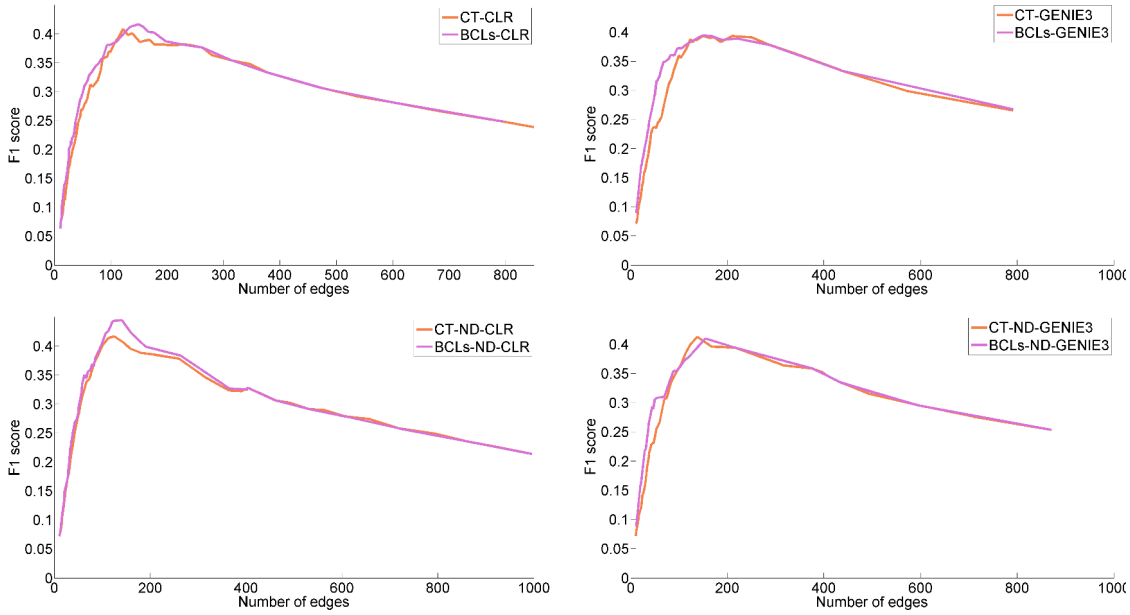


Figure 6.34 ~ *F*-PLOTS FOR THE DATASET 4 OF DREAM4 (*BRANE Clust-soft*) **~** *Curves depicting F-scores according to the number of edges (in a range from 10 to 1000), generated by CT or BRANE Clust on CLR, GENIE3, ND-CLR or ND-GENIE3 weights.*

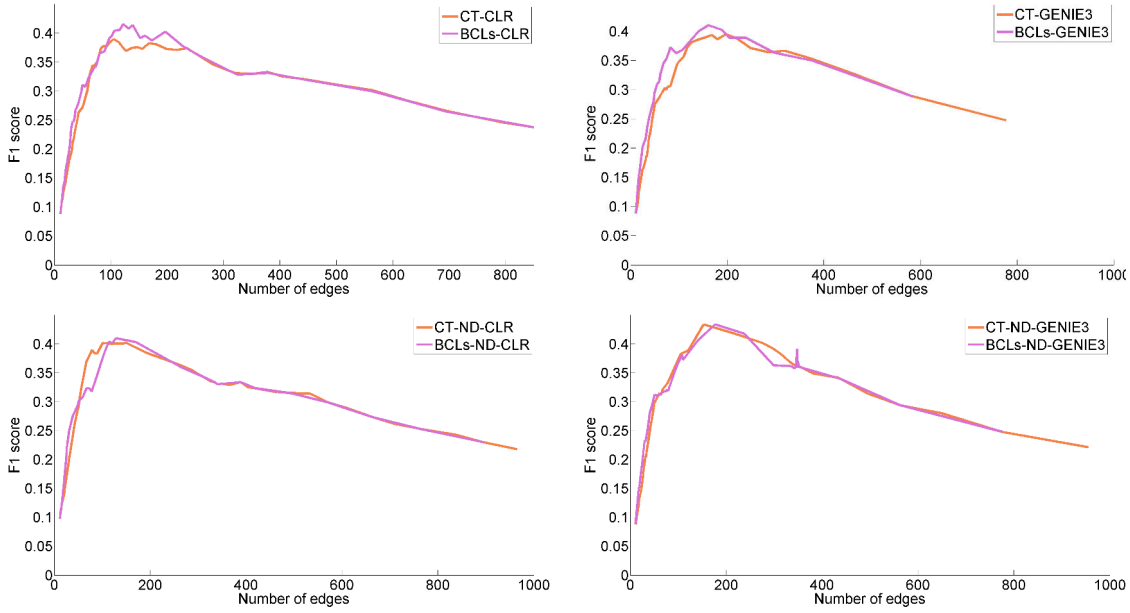


Figure 6.35 ~ *F*-PLOTS FOR THE DATASET 5 OF DREAM4 (*BRANE Clust-soft*) **~** *Curves depicting F-scores according to the number of edges (in a range from 10 to 1000), generated by CT or BRANE Clust on CLR, GENIE3, ND-CLR or ND-GENIE3 weights.*

7

Joint segmentation and restoration with higher-order graphical models (HOGMep)

"Essentially, all models are wrong, but some are useful."

George Edward Pelham Box

In this chapter, let us slide from graph inference to more generic data processing. The framework is related to non-blind inverse problems aiming at restoring a degraded signal from an observed one. In this work, we focus on multi-component signals. We consider each of them as a random variable, for which observations are available. In the HOGMep approach detailed in this chapter, a segmentation is jointly performed with the recovery. The Bayesian-based formulation is solved thanks to a Variational Bayesian Approximation (VBA). We firstly demonstrate the performance of HOGMep in an image deconvolution context by a comparison with state-of-the-art methods. HOGMep is then illustrated on an application example where cancer sufferers have to be distinguished. A promising performance is obtained, providing evidence for its potential interest for biological applications. This work is being consolidated in Pirayre et al. (2017).

Contents

7.1 Background on inverse problems

In this chapter, we recall the framework of inverse problems for which general information are first provided, before briefly presenting related works, specifically in the case of signal restoration and/or segmentation. Note that most of the notation is specific to this chapter and do not refer to previous notation employed in Chapters 4 to 6 in a GRN context, except for graphs, notably.

7.1.1 Importance of inverse problems

Inverse problems are largely encountered in signal, image and video processing (Pižurica *et al.*, 2004; Chaux *et al.*, 2007; Chaâri *et al.*, 2009), computer vision (Komodakis and Pesquet, 2015), medical imaging (Sonka and Fitzpatrick, 2000; Man *et al.*, 2001; Elbakri and Fessler, 2003), geophysics (Pham *et al.*, 2014; Repetti *et al.*, 2015), analytical chemistry (Ning *et al.*, 2014), microscopy (Dupé *et al.*, 2009; Jezierska *et al.*, 2012) or astronomy (Lantéri and Theys, 2005; Rodet *et al.*, 2008), to name a few. It implies four kinds of entities: a true but unknown signal $\mathbf{x} \in \mathbb{R}^N$, a degradation operator $\mathbf{H} \in \mathbb{R}^{M \times N}$, a noise $\mathbf{n} \in \mathbb{R}^M$ and finally a known but degraded signal $\mathbf{y} \in \mathbb{R}^M$ (also called observation). In such a case, inverse problems aim at recovering $\hat{\mathbf{x}} \in \mathbb{R}^N$ — an estimation of the true signal $\mathbf{x}$ — from knowledge on the observations $\mathbf{y}$ and the degradation operator $\mathbf{H}$. According to the degradation operator, signal recovery finds instances in denoising, debluring, segmentation or reconstruction problems. More conceptually, we want to recover information about a physical object from its measurements acquired by a given system. As sketched in Figure 7.1 in an image context, the usual linear model with additive noise links the true signal and the degraded one as follows:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{7.1}$$

Finding an estimator $\hat{\mathbf{x}}$ of the true signal $\mathbf{x}$ can be performed through two main approaches relying either on variational optimization or Bayesian strategy.

7.1.2 Methodologies for solving inverse problems

~ *Variational approach* ~ Assuming a Gaussian noise in (7.1) and $\mathbf{H}$ known[1], the estimator $\hat{\mathbf{x}}$ can be recovered by minimizing a data fidelity term, traditionally defined as a squared ℓ_2 norm related to the difference between the model $\mathbf{H}\mathbf{x}$ and the observations $\mathbf{y}$. However, according to Hadamard (1902), such a problem is said *ill-posed* as at least one of the existence, uniqueness and stability properties is violated. Regularization is thus used to restrict the realm of possibles by encoding constraints or *a priori* on the signal to be recovered — sparsity, positivity or bounding constraints for instance. The optimization problem can thus be expressed as

$$\underset{\mathbf{x} \in \mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2 \quad + \quad \lambda\, \phi(\mathbf{x}), \tag{7.2}$$

where ϕ is a function encoding the desired regularization and $\lambda > 0$, is a regularization parameter controlling the influence of the *a priori*. The choice of the regularization and the parameter

[1] The case where both $\mathbf{x}$ and $\mathbf{H}$ are unknown, referring to a blind inverse problem, is not addressed in this work.

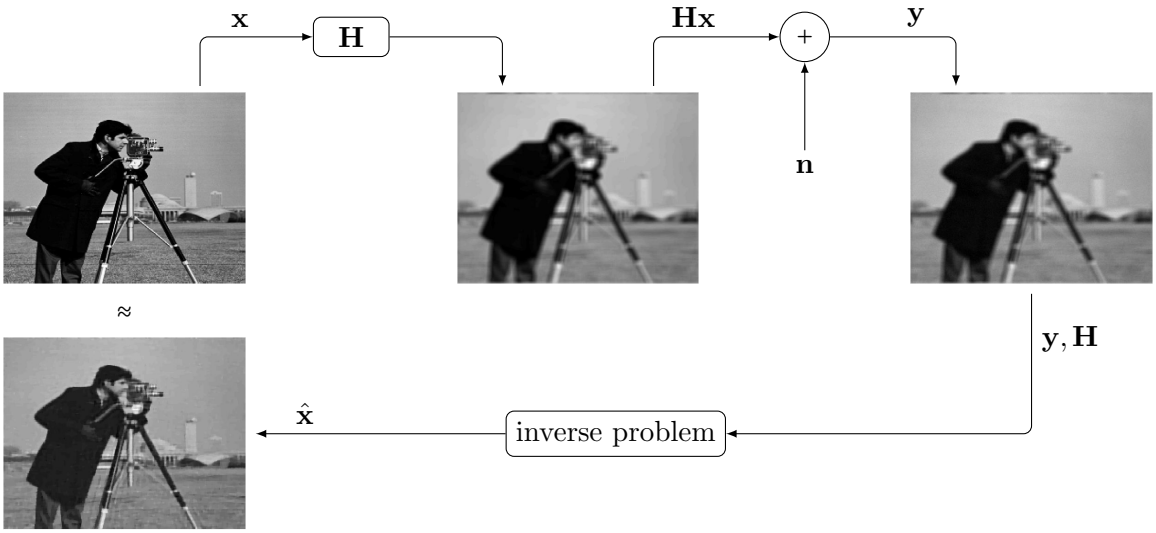


Figure 7.1 ~ SCHEME OF LINEAR MODELING WITH ADDITIVE NOISE ~
The physical object $\mathbf{x}$ *is firstly degraded by a blur operator* $\mathbf{H}$ *due to the image capture and an additive noise* $\mathbf{n}$ *linked to the acquisition is added to produce the degraded image* $\mathbf{y}$*. From* $\mathbf{y}$ *and* $\mathbf{H}$*, a (non blind) inverse problem aims at recovering an estimation* $\hat{\mathbf{x}}$ *of the true signal* $\mathbf{x}$*.*

controlling it is crucial and plays an important role in the quality of the estimation. The generic formulation in (7.2) is the root of a multitude of methods designed to improve the reconstruction of $\mathbf{x}$. Notably, among the usually used regularization, Tikhonov (Tikhonov, 1963) and sparsity-based are the most used. Sparsity-based regularization may rely on the ℓ_1 norm (Donoho *et al.*, 2006) or its variation such as the $\ell_2 - \ell_1$ like Huber criterion (Huber and Ronchetti, 2009), or the ℓ_2/ℓ_1 penalty (Zibulevsky and Pearlmutter, 2001), for instance. In addition, the Total Variation — local or non-local — (Rudin *et al.*, 1992; Gilboa and Osher, 2009) has proved its interest in an image processing context (Peyré, 2011; Chierchia *et al.*, 2014).

Nevertheless, inverse problems modeled in (7.1) also has a Bayesian interpretation, for which we provide some basics in the following. We deliberately detail some aspects as they are employed in our developed approach *HOGMep*.

~ Bayesian appproach ~ In a probabilistic context, both the signal to be recovered $\mathbf{x}$ and the observations $\mathbf{y}$ are assimilated to random variables. In such a case, we can assume, for each of them, the existence of a probability density function (pdf). The marginal pdf $p(\mathbf{x})$ encodes information about the signal to be recovered. It is chosen in order to reflect specific properties of the signal. The conditional pdf $p(\mathbf{y}\,|\,\mathbf{x})$ — termed likelihood of the observations — highlights the uncertainty present in the observations. It is driven by the underlying observation model e.g. (7.1) in our case.

An estimation of $\mathbf{x}$, can be determined from the knowledge of the posterior pdf $p(\mathbf{x}|\mathbf{y})$

(Bernardo and Smith, 1994). It reflects information about the signal to be recovered knowing the observations. Bayes' rule can thus be employed to obtain the posterior pdf:

$$p(\mathbf{x}\,|\,\mathbf{y}) = \frac{p(\mathbf{x})\,p(\mathbf{y}\,|\,\mathbf{x})}{p(\mathbf{y})}, \tag{7.3}$$

where $p(\mathbf{y}) = \int p(\mathbf{y}\,|\,\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$ is the marginal pdf of the observation. This term plays the role of a normalization constant of the posterior pdf, and turns out to be difficult to compute. In the following, we will see that, in practice, its computation can be avoided. From the posterior pdf, two kinds of estimators can be defined.

Maximum a posteriori (MAP) estimator The MAP estimator $\hat{\mathbf{x}}_{\mathrm{MAP}}$ is obtained by computing the mode of the posterior pdf:

$$\hat{\mathbf{x}}_{\mathrm{MAP}} = \arg\max_{\mathbf{x}} \; p(\mathbf{x}\,|\,\mathbf{y}). \tag{7.4}$$

Using (7.3), the MAP criterion is equivalent to

$$\hat{\mathbf{x}}_{\mathrm{MAP}} = \arg\max_{\mathbf{x}} \; \frac{p(\mathbf{x})\,p(\mathbf{y}\,|\,\mathbf{x})}{p(\mathbf{y})}. \tag{7.5}$$

As the denominator $p(\mathbf{y})$ does not depend on the variable $\mathbf{x}$, Problem (7.5) can be reduced to the maximization of the numerator of (7.5).

Is there a link with variational approaches? To answer this question, it can also be useful to consider the logarithm version of the MAP estimator. In such a case, the MAP criterion can be re-expressed as

$$\hat{\mathbf{x}}_{\mathrm{MAP}} = \arg\min_{\mathbf{x}} \; -\ln p(\mathbf{y}\,|\,\mathbf{x}) - \ln p(\mathbf{x}). \tag{7.6}$$

This formulation allows us to draw a parallel between variational approaches in Section 7.1.2 and the MAP criterion. Indeed, the first term in (7.6) is a data fidelity term while the second one refers to a regularization term. More specifically, it can be shown that, using the linear model with additive mean-zero Gaussian noise (7.1) and a suitable prior for $p(\mathbf{x})$, the MAP estimator can be determined as a solution to Problem (7.2). From a global view point, the MAP estimator can be computed through the minimization of a cost function. According to the properties of the cost function, we have at our disposal a large panel of algorithms. We can cite the most popular: descent algorithms, Expectation-Maximization (EM) (McLachlan and Krishnan, 2008), Majorize-Minimize (MM) strategy (Chouzenoux *et al.*, 2011), proximal algorithms (Combettes and Pesquet, 2011) or primal-dual methods (Chambolle and Pock, 2011; Komodakis and Pesquet, 2015). However, the MAP estimator is not the only one that can be used. We thus now present another classical estimator, usually named posterior mean.

Posterior Mean (PM) estimator The PM estimator $\hat{\mathbf{x}}_{\mathrm{PM}}$ is obtained by computing the mean of the posterior pdf:

$$\hat{\mathbf{x}}_{\mathrm{PM}} = \int \mathbf{x}\,p(\mathbf{x}\,|\,\mathbf{y})\mathrm{d}\mathbf{x}. \tag{7.7}$$

Unlike MAP, the PM estimator results in an integral computation for whose computation — in the large majority of case — is analytically intractable. The PM estimator can be obtained thanks to two main approaches classified into *i*) stochastic methods and *ii*) approximation methods. Briefly, the first one, referred to as Markov Chain Monte Carlo (MCMC), consists in generating a sufficiently large set of samples of i.i.d random variables from the desired distribution (Robert and Casella, 2004). The PM criterion is then determined as the empirical average over all these samples. For this purpose, the two most used MCMC algorithms are Metropolis-Hasting and the Gibbs sampler. The second one refers to methods providing an analytical approximation of the posterior pdf. While several approximations exist, we focus on the classical Variational Bayesian Approximation (VBA) (Parisi, 1998). As it is employed in our proposed method *HOGMep*, we dedicate Section 7.1.3 to theoretical aspects regarding VBA.

To go a little further... In a Bayesian framework, in addition to the variable of interest — the true signal, for instance — it is usual to estimate additional latent variables. This scheme allows us to introduce the concept of Bayesian hierarchical models (Molina, 1994), for which all notions previously introduced are valuable. Let us go back on the posterior pdf in (7.3). It involves — without taking account the normalization constant — the likelihood $p(\mathbf{y}\,|\,\mathbf{x})$ and the prior pdf $p(\mathbf{x})$, which can be respectively parametrized by hyperparameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Let us denote by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$, the set of hyperparameters following a prior distribution $p(\boldsymbol{\theta})$. This prior distribution, called hyperprior, can be parametrized by a set of parameters $\boldsymbol{\alpha}$. In such a case, a joint posterior pdf with respect to $\mathbf{x}$ and $\boldsymbol{\theta}$ can be defined:

$$p(\mathbf{x}, \boldsymbol{\theta}\,|\,\mathbf{y}) = \frac{p(\mathbf{y}\,|\,\mathbf{x}, \boldsymbol{\theta}_1)\, p(\mathbf{x}\,|\,\boldsymbol{\theta}_2)\, p(\boldsymbol{\theta})}{p(\mathbf{y})}. \tag{7.8}$$

This model, involving variables of interest and hyperparameters, is called a Bayesian hierarchical model. Classical MAP or PM estimators can thus be derived. However, due to its complicated form MAP, the estimator is not easily tractable. PM estimator is thus preferred and VBA can be employed to compute it.

7.1.3 Variational Bayesian Approximation theory

As mentioned, VBA strategy aims at providing an approximation of the true posterior pdf $p(\mathbf{x}|\mathbf{y})$. For this purpose, let us denote by $q(\mathbf{x})$ the approximated pdf. Our goal is to find a pdf as close as possible to the true pdf. This problem can be tackled by minimizing a dissimilarity measure between the approximated pdf $q(\mathbf{x})$ and the true one $p(\mathbf{x}\,|\,\mathbf{y})$. In a probabilistic context, the most intuitive dissimilarity measure is the Kullback-Leibler divergence ($\mathcal{KL}$) as it quantifies the difference between two pdfs. The optimal approximation $q^{\text{opt}}(\mathbf{x})$ can thus be obtained by solving the following optimization problem:

$$\begin{aligned} q^{\text{opt}}(\mathbf{x}) &= \arg\min_{q(\mathbf{x})} \; \mathcal{KL}(q(\mathbf{x}) \,\|\, p(\mathbf{x}\,|\,\mathbf{y})) \\ &= \arg\min_{q(\mathbf{x})} \; \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x}\,|\,\mathbf{y})} \mathrm{d}\mathbf{x}. \end{aligned} \tag{7.9}$$

Using conditional probability properties, we see that Problem (7.9) is equivalent to

$$q^{\text{opt}}(\mathbf{x}) = \arg\min_{q(\mathbf{x})} \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})p(\mathbf{y})}{p(\mathbf{x},\mathbf{y})} \mathrm{d}\mathbf{x}, \tag{7.10}$$

where $p(\mathbf{x},\mathbf{y})$ is the joint pdf, generally known. The integrand in (7.10) is classically decomposed into the sum of the logarithmic marginal pdf of the observations and the Gibbs free energy as follows:

$$q^{\text{opt}}(\mathbf{x}) = \arg\min_{q(\mathbf{x})} \ln p(\mathbf{y}) + \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x},\mathbf{y})} \mathrm{d}\mathbf{x}. \tag{7.11}$$

As the logarithmic marginal pdf of the observations $\log p(\mathbf{y})$ does not depend on $q(\mathbf{x})$, the optimization problem for finding the optimal approximation of $p(\mathbf{x}|\mathbf{y})$ is reduced to

$$q^{\text{opt}}(\mathbf{x}) = \arg\min_{q(\mathbf{x})} \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x},\mathbf{y})} \mathrm{d}\mathbf{x}. \tag{7.12}$$

Another trick has to be employed in order to avoid intractability due to mutual dependencies between variables to be estimated. For this purpose, let P be the number of variables to be estimated, and J an integer between 1 and P, the following separable distribution can be considered:

$$q(\mathbf{x}) = \prod_{j=1}^{J} q_j(\mathbf{x}_j), \tag{7.13}$$

where $(\mathbf{x}_j)_{1\leq j\leq J}$ represent disjoint subsets of $\mathbf{x}$ such that $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_J)$. Note that if $J = P$, the separability is total, otherwise we have a partial separability. Using a separable scheme for the variables to be estimated is equivalent to neglecting statistical links between them and simplify their computation. However, when the separability is total, a lack of correlation may become detrimental to the approximation. Although no general rule provides a choice in the level of separability, in practice it can be a compromise between the quality of the approximation and the level of simplification of the computation.

Anyway, taking this separability scheme into account — whatever its level — an explicit solution exists to Problem (7.9). Its is given, for all $j \in \{1, \ldots, J\}$, by

$$q_j^{\text{opt}}(\mathbf{x}_j) \propto \exp\left(\langle \ln p(\mathbf{y},\mathbf{x}) \rangle_{\prod_{i\neq j} q_i(\mathbf{x}_i)}\right), \tag{7.14}$$

where for any arbitrary variable $\mathbf{w}(x)$,

$$\langle \mathbf{w}(x) \rangle_{\prod_{i\neq j} q_i(\mathbf{x}_i)} = \int \mathbf{w}(x) \prod_{i\neq j} q_i(\mathbf{x}_i) \mathrm{d}\mathbf{x}_i, \tag{7.15}$$

which corresponds to the expectation of the variable $\mathbf{w}(x)$ with respect to the distribution of all unknown variables except the one of interest. Details can be found in Choudrey (2002) or Šmídl and Quinn (2006). Due to the implicit relations existing between pdfs $\left(q_j(\mathbf{x}_j)\right)_{1\leq j\leq J}$, an analytical expression of $q(\mathbf{x})$ does not exist generally. These distributions can thus be determined in an

iterative way, by updating one of the separable components $\left(q_j(\mathbf{x}_j)\right)_{1\leq j\leq J}$ while fixing the others.

Variational Bayesian Approximation (VBA) has been widely used in various applications such as in graphical model learning (Jordan *et al.*, 1999), image processing (Zheng *et al.*, 2015a), source separation (Choudrey, 2002) or super-resolution (Babacan *et al.*, 2011) to name a few.

At this point, Bayesian estimators and methods to compute them have been introduced. In Section 7.2, we will see how our proposed approach *HOGMep* tackles Bayesian hierarchical models and VBA for a joint restoration and segmentation on multi-component signals.

7.2 *HOGMep*: multi-component signal segmentation and restoration

7.2.1 Brief review on image segmentation and/or restoration

Although the proposed approach *HOGMep* can be applied to arbitrary multi-component signals for solving inverse problems, one of the most intuitive applications lies in image processing. We thus dedicate this section to a brief overview of a small portion of the huge literature (Cheng *et al.*, 2001) regarding image segmentation and/or restoration.

Image segmentation (or pixel clustering) aims at partitioning pixels into classes — spatially delimited by contours — sharing specific properties such as intensities or textures. For this purpose, Potts-Markov Random Fields (MRF) are traditionally used. Various strategies can be employed to solve the underlying problem such as convex optimization as in Komodakis *et al.* (2011) and Bioucas-Dias *et al.* (2014). In a Bayesian framework, the Iterated Conditional Modes (ICM) algorithm developed by Besag (1986) is one of the reference algorithms. Authors in Pereyra *et al.* (2012, 2013) used a Markov Chain Monte Carlo (MCMC) approach while a Variational Bayesian Approximation (VBA) is preferred by (McGrory *et al.*, 2009). Another strategy based on Variational Expectation-Maximization is proposed in Chaari *et al.* (2011). In the recent work by Pereyra and McLaughlin (2017), authors used a Potts model in a Bayesian framework and propose a novel strategy to estimate it while the regularization parameter of the model is automatically computed. Note that Potts-Markov random fields can be viewed as a Bayesian interpretation of energy functions solved by Graph cuts (Boykov *et al.*, 2001; Kolmogorov and Zabih, 2004). A similar interpretation can be drawn between continuous-valued MRF and combinatorial Dirichlet problem (Singaraju *et al.*, 2011) — used for image segmentation as in Grady (2006) and Sodjo *et al.* (2016) for instance. In Cai *et al.* (2013), image segmentation is performed *via* the Mumford-Shah model. In a different vein based on contour detection, watershed transformation (Beucher and Lantuéjoul, 1979) can also be considered as in Tarabalka *et al.* (2010) and Couprie *et al.* (2011), for instance.

Image restoration is a classical application of inverse problems where acquired images to be recovered are corrupted by a degradation operator (Pustelnik *et al.*, 2016). It can correspond to a blur during the acquisition or a projection operator as in tomography for instance. As for

segmentation, various strategies can be used. On the one hand, a variational approach can be used for solving image restoration problems. The quality of the results is mainly driven by the choice of the regularization terms. For instance, authors in Chouzenoux *et al.* (2013) propose the use of $\ell_2 - \ell_0$ functions and a Majorize-Minimize (MM) strategy for solving the underlying problem. In image processing application, the interest of a Total Variation (TV) regularization has been demonstrated (Chambolle and Pock, 2011; O'Connor and Vandenberghe, 2017). Improvement can be obtained using a Non-Local TV (NLTV) regularization as in Chierchia *et al.* (2014). The additional complexity of the cost function to be minimization can be solved using both proximal and primal-dual algorithms. In a multispectral images context, regularization can be defined in order to promote similarities between images (Briceño-Arias *et al.*, 2011). On the other hand, various Bayesian approaches have been proposed for image restoration.

On the other hand, in a Bayesian framework, a Gaussian prior for the image was traditionally used. While Molina *et al.* (1999) use a MAP estimator, an evidence approach is preferred in Babacan *et al.* (2010). Nevertheless, for estimating the joint posterior distribution, VBA is sometimes preferred. Indeed, in Likas and Galatsanos (2004); Chantas *et al.* (2008) and Chen *et al.* (2014), authors adapt VBA for (blind) image deconvolution. Complemental to Bayesian framework, wavelet transformation can be used for an image deconvolution purpose as in Figueiredo (2003) for instance. A prior on the wavelet coefficients of the image can be given through a Gaussian Scale Mixture (GSM). This choice is adopted in Bioucas-Dias (2006) where a generalized EM algorithm is used or in Portilla *et al.* (2003) in which the restored image is obtained *via* a least squares Bayesian estimator. As well-adapted to multi-component images, a Multivariate Exponential Power (MEP) distributions for the wavelet coefficients of the image is used in Marnissi *et al.* (2016). They propose to solve the resulting Bayesian problem using an MCMC strategy. Note that, as highlighted in Gómez-Sánchez-Manzano *et al.* (2008), GSM can be used to represent MEP distributions for particular shape parameter values.

However, instead of performing image restoration and segmentation in an independent manner, jointly proceeding can be considered and becomes trendy. Indeed, compared to the conventional segmentation, the joint restoration and segmentation is more robust to data degradations such as blur, noise[2]. We can notably evoke the work of Ayasso and Mohammad-Djafari (2010) where restoration and segmentation of single-component images are performed thanks to a VBA strategy applied on a hierarchical Bayesian modeling involving a Potts model for label variables and a Gaussian prior on pixels variables. Authors in Zhao *et al.* (2016a) develop a joint deconvolution and segmentation problem Bayesian method, based on a generalized Gaussian distribution and Potts model, for medical ultrasound images. A MCMC method is used to estimate the unknown parameters. Other approaches consider variational formulations. Indeed, a fuzzy c-means functional penalized by a TV regularizer is proposed by He *et al.* (2012), and for which an ADMM (Alternating Direction Method of Multipliers) algorithm is used. In Paul *et al.* (2013), authors propose to model joint segmentation and restoration through generalized linear models and Bregman divergence for which an alternating minimization algorithm is used.

[2]Note than we followed a similar philosophy in *BRANE Clust* (Chapter 6), with joint inference and clustering. Better integrating heterogeneous processing steps reduces artifacts caused by motley input/output model assumptions.

However, this approach is restricted to a binary segmentation of single-component images. To close this brief review, we evoke the work of Cai (2015) developed to perform segmentation of multi-component images by integrating image restoration framework in their model. The variational formulation they propose is based on the Mumford-Shah model and a TV regularization, two famous models borrowed from image segmentation and restoration fields. An alternating minimization algorithm is used to solve the underlying problem.

We now detail the proposed approach, named *HOGMep*, for joint restoration and segmentation tasks performed on multi-component signals.

7.2.2 Inverse problem formulation and priors

As mentioned in Section 7.1.1, we concentrate on the standard inverse problem consisting of recovering an unknown signal $\mathbf{x}$ from a degraded one $\mathbf{y}$. We thus consider the linear model with additive noise formulated in (7.1). In our approach, we are interested in B-component signals (Chaux *et al.*, 2008, 2009) where $\mathbf{x} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_N^\top]^\top$ and, for every variable $i \in \{1, \dots, N\}$, $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,B})^\top$. We thus define, for $(M, N, B) \in (\mathbb{N}^*)^3$, $\mathbf{y} \in \mathbb{R}^M$ as the observed data, $\mathbf{x} \in \mathbb{R}^{NB}$ the unknown signal to be recovered, $\mathbf{H} \in \mathbb{R}^{M \times NB}$ a linear degradation operator and $\mathbf{n}$ as a noise, supposed statistically independent of $\mathbf{x}$. The model now defined, we focus in the following on the choice of prior distributions.

~ *Likelihood prior* **~** We recall that the likelihood corresponds to the distribution of the observations given the data. Its definition is driven with the observation model given in (7.1). Assuming a zero-mean white Gaussian noise with inverse variance γ, the desired likelihood $p(\mathbf{y} \,|\, \mathbf{x}, \gamma)$ can be modeled as a Normal distribution with mean $\mathbf{Hx}$ and covariance matrix $\gamma^{-1}\mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix. More formally, we have:

$$p(\mathbf{y} \,|\, \mathbf{x}, \gamma) = \mathcal{N}(\mathbf{Hx}, \gamma^{-1}\mathbf{I}). \tag{7.16}$$

Note that γ plays the role of an hyperparameter and we provide additional details later in this section for additional definition of it.

As previously mentioned, the Bayesian framework requires a prior $p(\mathbf{x})$, the distribution of the desired signal $\mathbf{x}$. However, as our objective is to perform, in conjunction with the recovery task, a classification of the components of $\mathbf{x}$, we have to introduce a label field. For this purpose, L being the number of expected classes, the label field is encoded by a vector of hidden variables $\mathbf{z} \in \{1, \dots, L\}^N$ with a distribution $p(\mathbf{z})$. In such a case, the prior on $\mathbf{x}$ becomes dependent on the class i.e. according to the value on $\mathbf{z}$, the probability on $\mathbf{x}$ may change. As a result, we have to define a prior for $p(\mathbf{x} \,|\, \mathbf{z})$, the conditional distribution of $\mathbf{x}$ given the hidden variable $\mathbf{z}$. Let us now detail the chosen prior associated to the hidden variables $\mathbf{z}$ and the signal $\mathbf{x}$ — in a given class — we want to estimate.

~ *Sought data prior* **~** A usual way to estimate the hidden variables $\mathbf{z}$ is to use a Potts model on $\mathbf{z}$. This model can be defined on a general graph structure $\mathrm{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of

nodes and $\mathcal{E}$ the set of edges. For each node i in $\mathcal{V}$, a discrete variable z_i taking its value among L distinct values can be defined. The distribution $p(\mathbf{z})$ associated to such a model is given by

$$p(\mathbf{z}) \propto \exp\left(\frac{\beta}{2}\sum_{i=1}^{N}\sum_{j\in\mathcal{V}(i)}\delta(z_i,z_j)\right), \tag{7.17}$$

where $\mathcal{V}(i)$ is the set of indices for the neighbors of x_i, δ is the Kronecker delta function taking 1 if $z_i = z_j$ and 0 otherwise, and β is the Potts parameter. This model has been widely used in image processing for segmentation purposes (McGrory *et al.*, 2009; Ayasso and Mohammad-Djafari, 2010; Bioucas-Dias *et al.*, 2014; Pereyra and McLaughlin, 2017).

Nevertheless, the main limitation of the Potts model is its restriction to pairwise interaction between variables. To overcome this, arbitrary Higher-Order Graphical Models (HOGM) can be used (Marinari and Marra, 1990; Zheleva *et al.*, 2010). They extend the Potts model to cliques of arbitrary size. In such a case, the distribution $p(\mathbf{z})$ becomes

$$p(\mathbf{z}) \propto \exp\left(\sum_{s=1}^{S}\sum_{(i_1,\dots,i_s)\in\mathcal{N}_s} V_s(z_{i_1},\dots,z_{i_s})\right), \tag{7.18}$$

where S is the size of the maximal clique and, for every $s \in \{1,\dots,S\}$, the function V_s is a potential function of order s, and $\mathcal{N}_s$ is the set of cliques of size s. The model contains a prior weighting parameter $\boldsymbol{\lambda}$, not explicitly written in (7.18).

In addition to a prior on hidden label variables, a conditional distribution of $\mathbf{x}$ given label variables $\mathbf{z}$ has to be assumed. While a Gaussian one can be employed (Ayasso and Mohammad-Djafari, 2010), the MEP distribution introduced in Gómez *et al.* (1998), denoted by $\mathcal{M}$, is preferred in *HOGMep*. Given an r-dimensional random variable $\mathbf{w}$, the MEP pdf is given, for every $\mathbf{w} \in \mathbb{R}^r$, by

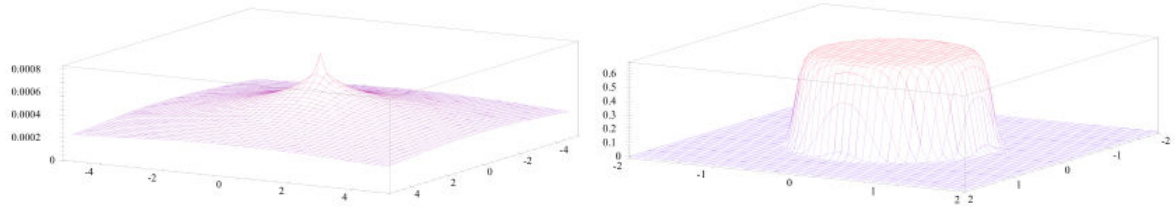
$$\mathcal{M}(\mathbf{w};\mathbf{m},\boldsymbol{\Omega},\beta) = \kappa|\boldsymbol{\Omega}|^{\frac{1}{2}}\exp\left(-\frac{1}{2}\left((\mathbf{w}-\mathbf{m})^{\top}\boldsymbol{\Omega}(\mathbf{w}-\mathbf{m})\right)^{\beta}\right), \tag{7.19}$$

where

$$\kappa = \frac{r\Gamma\left(\frac{r}{2}\right)}{\pi^{\frac{r}{2}}\,\Gamma\left(1+\frac{r}{2\beta}\right)2^{1+\frac{r}{2\beta}}}, \tag{7.20}$$

and Γ is the gamma function, $\boldsymbol{\Omega} \in \mathbb{R}^{r\times r}$ is a symmetric positive definite matrix, $\mathbf{m} \in \mathbb{R}^r$, and $\beta > 0$ is the exponent determining the shape of the distribution. Illustrations for two different shape parameters are displayed in Figure 7.2. Such a prior is well suited to multi-component images (Marnissi *et al.*, 2016). Note that setting the shape parameter β to 1 reduces the MEP distribution to a Gaussian one.

Assuming a MEP distribution for $\mathbf{x}$ conditionally to $\mathbf{z}$ means that for every class labeled by $l \in \{1,\dots,L\}$, variables $\mathbf{x}_i$ belonging to this class — in other words, variables $\mathbf{x}_i$ having a label



(a) Density function with $\beta = 0.25$. (b) Density function with $\beta = 10$.

Figure 7.2 ~ MULTIVARIATE EXPONENTIAL POWER (MEP) PDFS ~
Probability density functions for a MEP distribution with (a) $\beta = 0.25$ and (b) $\beta = 10$. Illustrations are reproduced from Gómez et al. (1998).

vector z_i equal to l, for all $i \in \{1, \dots, N\}$ — follow a MEP distribution with parameters $\mathbf{m}_l$, $\mathbf{\Omega}_l$ and β_l:

$$p(\mathbf{x}_i \,|\, \mathbf{z}_i = l, \mathbf{m}, \mathbf{\Omega}, \boldsymbol{\beta}) \;=\; \mathcal{M}(\mathbf{x}_i; \mathbf{m}_l, \mathbf{\Omega}_l, \beta_l), \tag{7.21}$$

where $\mathbf{m} = [\mathbf{m}_1, \dots, \mathbf{m}_L]^\top$, $\mathbf{\Omega} = [\mathbf{\Omega}_1, \dots, \mathbf{\Omega}_L]$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_L)^\top$ contain the parameters of the MEP distributions associated with the L label values. As a result, we have that the conditional distribution of $\mathbf{x}$ given the label variables $\mathbf{z}$ is:

$$p(\mathbf{x} \,|\, \mathbf{z}, \mathbf{m}, \mathbf{\Omega}, \boldsymbol{\beta}) = \prod_{i=1}^{N} p(\mathbf{x}_i \,|\, z_i, \mathbf{m}, \mathbf{\Omega}, \boldsymbol{\beta}). \tag{7.22}$$

Note that, as it is the case for the likelihood previously introduced, some hyper-parameters appear. Specifically, there are three hyperparameters: $\mathbf{m}$, $\mathbf{\Omega}$ and $\boldsymbol{\beta}$ and we will see in the later section how to integrate them in the global model.

At this point, all three required distributions are defined — $p(\mathbf{y} \,|\, \mathbf{x}, \gamma)$, $p(\mathbf{z})$ and $p(\mathbf{x} \,|\, \mathbf{z})$. Nevertheless, for all $l \in \{1, \dots, L\}$, restricting the shape parameter β_l to the interval $(0, 1]$, the MEP distribution can be represented as Gaussian Scale Mixtures (GSM) (Gómez-Sánchez-Manzano *et al.*, 2008) i.e. the integral of Gaussian distributions with a fixed mean $\mathbf{m}_l$ and a variable variance $u_i^{-1}\mathbf{\Omega}_l^{-1}$:

$$(\forall l \in \{1, \dots, L\}) \quad \mathcal{M}(\mathbf{x}_i; \mathbf{m}_l, \mathbf{\Omega}_l, \beta_l) = \int_{\mathbb{R}^+} \mathcal{N}(\mathbf{x}_i; \mathbf{m}_l, u_i^{-1}\mathbf{\Omega}_l^{-1}) p(u_i \,|\, \beta_l) \mathrm{d}u_i, \tag{7.23}$$

where u_i is assimilated to a latent variable, for which the pdf given the shape parameter β_l is denoted by $p(u_i \,|\, \beta_l)$. When $\beta_l < 1$, this pdf can be expressed as a function of a positive alpha-stable distribution. When $\beta_l = 1$, it degenerates into a Dirac distribution (Gómez-Sánchez-Manzano *et al.*, 2008). Note that the pdf of an apha-stable distribution cannot be generally expressed in a closed form. However, we will see in Section 7.2.3 how our approach allows us to circumvent this difficulty. For the following, let $\mathbf{u} = (u_1, \dots, u_N)^\top$ be a vector gathering all introduced latent variables.

Likelihood and model prior distributions now defined, we have to deal with the introduced hyperparameters.

~ *Hyperpriors* ~ Our proposed Bayesian formulation involves four hyperparameters: the inverse noise variance γ, the mean variables $(\mathbf{m}_l)_{1\leq l\leq L}$, the inverse covariance matrices $(\mathbf{\Omega}_l)_{1\leq l\leq L}$ and the shape parameters $(\beta_l)_{1\leq l\leq L}$. While authors in Wand *et al.* (2010) estimate shape parameters by assigning an uniform distributions to them, in our work, we will restrict our attention to the case when all the shape parameters of the MEP distributions are identical, i.e. $\beta_1 = \cdots = \beta_L = \beta$. In practice, this single parameter is thus fixed in advance, according to our prior knowledge. It remains to assign hyperpriors to the three leftover hyperparameters. Let $\mathcal{G}$ and $\mathcal{W}$ denote Gamma and Wishart distributions, respectively, we assume that:

$$\begin{aligned} p(\gamma) &= \mathcal{G}(\bar{a}, \bar{b}), \\ p(\mathbf{m}_l) &= \mathcal{N}(\bar{\boldsymbol{\mu}}, \bar{\mathbf{\Lambda}}), \\ p(\mathbf{\Omega}_l) &= \mathcal{W}(\bar{\mathbf{\Gamma}}, \bar{\nu}). \end{aligned} \tag{7.24}$$

We obtain a Bayesian hierarchical model — named *HOGMep* — for which dependency relationships between the variables are summarized in Figure 7.3.

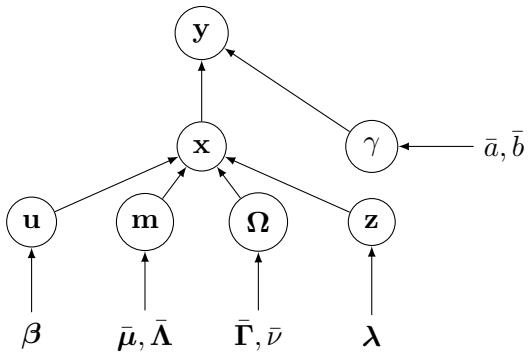


Figure 7.3 **~** DEPENDENCY RELATIONSHIPS BETWEEN VARIABLES IN *HOGMep* **~**

To the best of our knowledge, HOGM and MEP priors have not been jointly used for image recovery and segmentation tasks. As all requirements for Bayesian inference are established, we now present how to define the corresponding joint posterior distribution before detailing the VBA strategy used to approximate it.

~ *Joint posterior distribution* ~ As mentioned in Section 7.1.2, the estimation of the unknown signal to be recovered is obtained through the posterior distribution. In our model, not only the signal $\mathbf{x}$ has to be estimated. Indeed, in addition to $\mathbf{x}$, latent variables $\mathbf{z}$ and $\mathbf{u}$ have to be estimated as well as hyperparameters γ, $\mathbf{m}$ and $\mathbf{\Omega}$. For this purpose, their joint estimation can be obtained thanks to a joint probability distribution. The latter was defined using Baye's rule, leading to $p(\mathbf{x}, \mathbf{u}, \mathbf{z}, \gamma, \mathbf{m}, \mathbf{\Omega} \,|\, \mathbf{y}, \beta)$ proportional to

$$p(\mathbf{y} \,|\, \mathbf{x}, \gamma) \prod_{i=1}^{N} \Big(p(\mathbf{x}_i \,|\, z_i, u_i, \mathbf{m}, \mathbf{\Omega}) p(u_i \,|\, \beta) \Big) p(\mathbf{z}) p(\gamma) \prod_{l=1}^{L} p(\mathbf{m}_l) p(\mathbf{\Omega}_l) \tag{7.25}$$

This posterior distribution has an intricate form due to the dependence between the unknown variables. To tackle this problem, two approaches can be mainly employed: MCMC approaches (Pereyra *et al.*, 2013) and Variational Bayesian Approximation (McGrory *et al.*, 2009; Ayasso and Mohammad-Djafari, 2010; Chaari *et al.*, 2011). We thus now detail how VBA can lead to an elegant solution.

7.2.3 Variational Bayesian Approximation and algorithm

In order to use a VBA strategy, we introduce, in the following, a vector $\Theta = (\Theta_j)_{1\leq j\leq J}$ where all variables $(\mathbf{x}, \mathbf{u}, \mathbf{z}, \gamma, \mathbf{m}, \mathbf{\Omega})$ which will be estimated are stored.

~ *VBA of the HOGMep model* ~ Going back over VBA theory introduced in Section 7.1.3, we aim at finding a pdf $q(\Theta)$ which approximates the true posterior distribution $p(\Theta\,|\,\mathbf{y})$ by minimizing the following Kullback-Leibler ($\mathcal{KL}$) divergence between $q(\Theta)$ and $p(\Theta\,|\,\mathbf{y})$

$$\mathcal{KL}\left(q(\Theta)||p(\Theta\,|\,\mathbf{y})\right) = \int q(\Theta)\ln\frac{q(\Theta)}{p(\Theta\,|\,\mathbf{y})}\mathrm{d}\Theta. \tag{7.26}$$

Here, we allow variable Θ_j, with $j \in \{1,\ldots,J\}$ to be either continuous or discrete by replacing the integral with a sum if required. As already pointed out, the optimal approximate distribution can be computed from the following expression (Šmídl and Quinn, 2006):

$$(\forall j \in \{1,\ldots,J\}) \quad q(\Theta_j) \propto \exp\left(\langle \ln p(\mathbf{y},\Theta)\rangle_{q_{-\Theta_j}}\right), \tag{7.27}$$

where $q_{-\Theta_j} = \prod_{i\neq j} q(\Theta_i)$ and $\langle\cdot\rangle_q$ denotes the expectation with respect to a probability distribution q. We have thus

$$\langle \ln p(\mathbf{y},\Theta)\rangle_{q_{-\Theta_j}} = \int \ln p(\mathbf{y},\Theta)\prod_{i\neq j} q(\Theta_i)\mathrm{d}\Theta_i. \tag{7.28}$$

Implicit relations between pdfs $(q(\Theta_j))_{1\leq j\leq J}$ generally prevent analytical expressions for $q(\Theta)$. Most frequently, these distributions are determined in an iterative way, by updating one of the separable components $\left(q(\Theta_j)\right)_{1\leq j\leq J}$ while fixing the others. Hence, to apply the VBA, the first step is to specify our separability assumptions. In this work, we consider the following separable form for the approximation:

$$q(\Theta) = \prod_{i=1}^{N}\left(q(\mathbf{x}_i, z_i)q(u_i)\right) q(\gamma)\prod_{l=1}^{L}\left(q(\mathbf{m}_l)q(\mathbf{\Omega}_l)\right), \tag{7.29}$$

with $q(\mathbf{x}_i, z_i) = q(\mathbf{x}_i|z_i)q(z_i)$. Hence, using (7.27), for every $i \in \{1,\ldots,N\}$ and $l \in \{1,\ldots,L\}$, the optimal solutions for $q(\mathbf{x}_i|z_i)$, $q(z_i)$, $q(\mathbf{m}_l)$, $q(\mathbf{\Omega}_l)$ and $q(\gamma)$ are such that

$$\begin{aligned} q(\mathbf{x}_i|z_i = l) &= \mathcal{N}(\boldsymbol{\eta}_{i,l}, \mathbf{\Xi}_{i,l}),\\ q(z_i = l) &= \pi_{i,l},\\ q(\mathbf{m}_l) &= \mathcal{N}(\boldsymbol{\mu}_l, \mathbf{\Lambda}_l),\\ q(\mathbf{\Omega}_l) &= \mathcal{W}(\mathbf{\Gamma}_l, \nu_l),\\ q(\gamma) &= \mathcal{G}(a, b). \end{aligned} \tag{7.30}$$

Since these distributions belong to known parametrized families of distributions, their optimization can be performed by iteratively updating their parameters. In the following, assuming that $k \in \mathbb{N}$ designates the iteration number, we describe how to estimate iteratively these distributions by deriving closed form expressions for their parameters.

~ *Determination of model pdf* $q(\mathbf{x}_i, z_i)$ ~ According to (7.25) and (7.27), the approximation of $q(\mathbf{x}_i, z_i)$ at iteration $k+1$ reads

$$\begin{aligned} q^{k+1}(\mathbf{x}_i, z_i) &= q^{k+1}(\mathbf{x}_i \,|\, z_i) q^{k+1}(z_i) \\ &\propto \exp\left(\left\langle \ln p(\mathbf{y} \,|\, \mathbf{x}, \gamma) \;+\; \sum_{j=1}^{N} \ln p(x_j \,|\, u_j, z_j, \mathbf{m}_{zj}, \mathbf{\Omega}_{zj}) \;+\; \ln p(\mathbf{z}) \right\rangle_{q^k_{-(x_i,z_i)}} \right). \end{aligned} \tag{7.31}$$

As mentioned in (7.30), $q^{k+1}(\mathbf{x}_i \,|\, z_i = l)$ is a Gaussian distribution whose covariance matrix $\mathbf{\Xi}_{i,l}^{k+1}$ and mean $\boldsymbol{\eta}_{i,l}^{k+1}$ are given, at iteration $k+1$, by

$$\mathbf{\Xi}_{i,l}^{k+1} = \left(\widehat{\gamma}^k \mathbf{H}_i^\top \mathbf{H}_i \;+\; \widehat{u}_i^k \widehat{\mathbf{\Omega}}_l^k \right)^{-1}, \tag{7.32}$$

$$\boldsymbol{\eta}_{i,l}^{k+1} = \mathbf{\Xi}_{i,l}^{k+1} \left(\widehat{\gamma}^k \mathbf{H}_i^\top (\mathbf{y} \;-\; \sum_{j<i} \mathbf{H}_j \widehat{\mathbf{x}}_j^{k+1} \;-\; \sum_{j>i} \mathbf{H}_j \widehat{\mathbf{x}}_j^k) \;+\; \widehat{u}_i^k \widehat{\mathbf{\Omega}}_l^k \boldsymbol{\mu}_l^k \right). \tag{7.33}$$

In the above expressions, $\mathbf{H}$ has been decomposed columnwise as $[\mathbf{H}_1, \ldots, \mathbf{H}_N]^\top$, where for every $i \in \{1, \ldots, N\}$, $\mathbf{H}_i \in \mathbb{R}^{M \times B}$. Furthermore, for an arbitrary variable $\mathbf{w}$, $\widehat{\mathbf{w}}^k$ is its expectation at iteration k.

We can then derive the expressions of the probabilities $q^{k+1}(z_i = l)$:

$$\begin{aligned} q^{k+1}(z_i = l) &= \pi_{i,l}^{k+1} \\ &\propto \left(|\mathbf{\Xi}_{i,l}^{k+1}| |\mathbf{\Gamma}_l^k| \right)^{1/2} \exp\left(\frac{1}{2} (\boldsymbol{\eta}_{\boldsymbol{i,l}}^{\boldsymbol{k+1}})^\top (\mathbf{\Xi}_{i,l}^{k+1})^{-1} \boldsymbol{\eta}_{i,l}^{k+1} \;+\; \frac{1}{2} \sum_{b=1}^{B} \psi\left(\frac{\nu_l^k + 1 - b}{2} \right) \right. \\ &\quad \left. -\; \frac{1}{2} \widehat{u}_i^k \mathrm{tr}[(\mathbf{\Lambda}_l^k + \boldsymbol{\mu}_l^k (\boldsymbol{\mu}_l^k)^\top) \widehat{\mathbf{\Omega}}_l^k] \;+\; \tilde{V}_l^k \right), \end{aligned} \tag{7.34}$$

where ψ is the digamma function[3] and

$$\widehat{V}_l^k = V_1(l) + \sum_{s=1}^{S-1} \left\langle \sum V_{s+1}(l, z_{i_1}, \ldots, z_{i_s}) \right\rangle_{\prod_{j \neq i} q^k(z_j)}.$$

Then, it can be noticed that, for every $j \in \{1, \ldots, N\}$,

$$\widehat{\mathbf{x}}_j^{k+1} = \sum_{l=1}^{L} \pi_{j,l}^{k+1} \eta_{j,l}^{k+1}. \tag{7.35}$$

The expression of $q^{k+1}(u_i)$ is derived from (7.25) and (7.27):

$$q^{k+1}(u_i) \;\propto\; \exp\left(\langle \ln p(\mathbf{x}_i | u_i, z_i, \mathbf{m}, \mathbf{\Omega}) \;+\; \ln p(u_i | \beta) \rangle_{q^k_{-u_i}} \right). \tag{7.36}$$

[3] The logarithmic derivative of the Gamma function.

However, the above expression does not have an analytical expression due to the lack of closed form expression for $p(u_i\,|\,\beta)$. Thus, instead of computing an analytical expression for $q(u_i)$, we focus on the expectation of u_i. As a MEP prior distribution can be expressed as a GSM, the integral form of the latter allows us to express the mean value of u_i at iteration $k+1$ as follows (Palmer *et al.*, 2005; Zheng *et al.*, 2015b):

$$\widehat{u}_i^{k+1} = \beta \left(\sum_{l=1}^{L} \pi_{i,l}^{k+1} \mathrm{tr}[\mathbf{A}_k \widehat{\boldsymbol{\Omega}}_l^k] \right)^{\beta-1}, \tag{7.37}$$

where $\mathbf{A}_k = (\boldsymbol{\eta}_{i,l}^{k+1} - \boldsymbol{\mu}_l^k)(\boldsymbol{\eta}_{i,l}^{k+1} - \boldsymbol{\mu}_l^k)^\top + \boldsymbol{\Xi}_{i,l}^{k+1} + \boldsymbol{\Lambda}_l^k$. This allows us to derive an approximate distribution of $\mathbf{x}_i$, which depends on the mean value of u_i.

~ *Determination of hyperpriors pdfs $q(\mathbf{m}_l)$, $q(\boldsymbol{\Omega}_l)$ and $q(\gamma)$* ~ According to (7.25) and (7.27), the approximation of $q^{k+1}(\mathbf{m}_l)$ at iteration $k+1$ reads:

$$q^{k+1}(\mathbf{m}_l) \propto \exp\left(\langle \ln p(\mathbf{x}|\mathbf{u}, \mathbf{z}, \mathbf{m}, \boldsymbol{\Omega}) + \ln p(\mathbf{m}) \rangle_{q^k_{-\mathbf{m}_l}} \right). \tag{7.38}$$

As mentioned in (7.30), $q^{k+1}(\mathbf{m}_l)$ is a Gaussian distribution for which the covariance matrix $\boldsymbol{\Lambda}_l^{k+1}$ and the mean $\boldsymbol{\mu}_l^{k+1}$ are given, at the iteration $k+1$, by

$$\boldsymbol{\Lambda}_l^{k+1} = \left(\bar{\boldsymbol{\Lambda}}^{-1} + \widehat{\boldsymbol{\Omega}}_l^k \sum_{i=1}^{N} \pi_{i,l}^{k+1} \widehat{u}_i^{k+1} \right)^{-1}, \tag{7.39}$$

$$\boldsymbol{\mu}_l^{k+1} = \boldsymbol{\Lambda}_l^{k+1} \left(\bar{\boldsymbol{\Lambda}}^{-1} \bar{\boldsymbol{\mu}} + \widehat{\boldsymbol{\Omega}}_l^k \sum_{i=1}^{N} \pi_{i,l}^{k+1} \widehat{u}_i^{k+1} \boldsymbol{\eta}_{i,l}^{k+1} \right). \tag{7.40}$$

Then, the mean value of $q^{k+1}(\mathbf{m}_l)$ is $\widehat{\mathbf{m}}_l^{k+1} = \boldsymbol{\mu}_l^{k+1}$.

From (7.25) and (7.27), we can derive the expression of $q^{k+1}(\boldsymbol{\Omega}_l)$ at iteration $k+1$ as follows:

$$q^{k+1}(\boldsymbol{\Omega}_l) \propto \exp\left(\langle \ln p(\mathbf{x}|\mathbf{u}, \mathbf{z}, \mathbf{m}, \boldsymbol{\Omega}) + \ln p(\boldsymbol{\Omega}) \rangle_{q^k_{-\boldsymbol{\Omega}_l}} \right). \tag{7.41}$$

Based on (7.30), $q^{k+1}(\boldsymbol{\Omega}_l)$ is a Wishart distribution parametrized by

$$\nu_l^{k+1} = \bar{\nu} + \sum_{i=1}^{N} \pi_{i,l}^{k+1}, \tag{7.42}$$

$$\boldsymbol{\Gamma}_l^{k+1} = \left(\sum_{i=1}^{N} \pi_{i,l}^{k+1} \widehat{u}_i^{k+1} \tilde{\mathbf{A}}_k + \bar{\boldsymbol{\Gamma}}^{-1} \right)^{-1}, \tag{7.43}$$

where $\tilde{\mathbf{A}}_k = (\boldsymbol{\eta}_{i,l}^{k+1} - \boldsymbol{\mu}_l^{k+1})(\boldsymbol{\eta}_{i,l}^{k+1} - \boldsymbol{\mu}_l^{k+1})^\top + \boldsymbol{\Xi}_{i,l}^{k+1} + \boldsymbol{\Lambda}_l^{k+1}$ and the mean value of $q^{k+1}(\boldsymbol{\Omega}_l)$ is

$$\widehat{\boldsymbol{\Omega}}_l^{k+1} = \nu_l^{k+1} \boldsymbol{\Gamma}_l^{k+1}. \tag{7.44}$$

Finally, the $q^{k+1}(\gamma)$ distribution is a Gamma distribution with parameters

$$a^{k+1} = \bar{a} + \frac{M}{2} \tag{7.45}$$

$$b^{k+1} = \bar{b} + \frac{1}{2}\|\mathbf{y}-\mathbf{H}\widehat{\mathbf{x}}^{k+1}\|^2 + \sum_{i=1}^{N}\sum_{l=1}^{L}\pi_{i,l}^{k+1}\mathrm{tr}[\mathbf{H}_i^\top\mathbf{H}_i\boldsymbol{\Xi}_{i,l}^{k+1}]. \tag{7.46}$$

From standard properties of the Gamma distribution, the expectation of γ at the iteration $k+1$ is equal to

$$\widehat{\gamma}^{k+1} = \frac{a^{k+1}}{b^{k+1}} = \frac{1}{b^{k+1}}\left(\bar{a} + \frac{M}{2}\right). \tag{7.47}$$

~ *Resulting algorithm* ~ Altogether, our algorithm can be summed up as follows:

Algorithm 4: *HOGMep*

Set initial values: $\boldsymbol{\eta}_{i,l}^0$, $\boldsymbol{\Xi}_{i,l}^0$, $\widehat{u}_i^0$, $\pi_{i,l}^0$, $\boldsymbol{\mu}_l^0$, $\boldsymbol{\Lambda}_l^0$, $\boldsymbol{\Gamma}_l^0$, ν_l^0, b^0, and set $a^k \equiv \bar{a} + \frac{M}{2}$;
Compute $\widehat{\boldsymbol{x}}_i^0 = \sum_{l=1}^{L}\pi_{i,l}^0\eta_{i,l}^0$, $\widehat{\boldsymbol{\Omega}}_l^0 = \nu_l^0\boldsymbol{\Gamma}_l^0$, and $\widehat{\gamma}^0 = 1/b^0(\bar{a} + M/2)$;
for *k = 0, 1, ...* **do**
  Update parameters $\boldsymbol{\Xi}_{i,l}^{k+1}$ and $\boldsymbol{\eta}_{i,l}^{k+1}$ of $q^{k+1}(\boldsymbol{x}_i|z_i = l)$ using (7.32) and (7.33).
  Compute $\pi_{i,l}^{k+1}$ from (7.34);
  Update mean values $\widehat{u}_i^{k+1}$ of $q^{k+1}(u_i)$ using (7.37);
  Update parameters $\boldsymbol{\Lambda}_l^{k+1}$ and $\boldsymbol{\mu}_l^{k+1}$ of $q^{k+1}(\mathbf{m}_l)$ using (7.39) and (7.40);
  Update parameters ν_l^{k+1} and $\boldsymbol{\Gamma}_l^{k+1}$ of $q^{k+1}(\boldsymbol{\Omega}_l)$ using (7.42) and (7.43). Compute $\widehat{\boldsymbol{\Omega}}_l^{k+1}$ from (7.44);
  Update parameter b^{k+1} of $q^{k+1}(\gamma)$ using (7.46). Compute $\widehat{\gamma}^{k+1}$ from (7.47).

We now present the practical interest of *HOGMep* in both an image processing context and a biological application. In all our simulations, *HOGMep* is used to estimate the unknown data, its classification as well as the noise level.

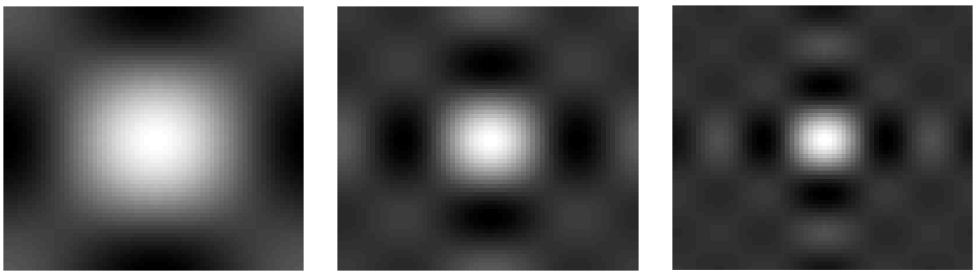
7.3 *HOGMep*: application to image processing and biological data

The performance of *HOGMep* is firstly assessed through segmentation and recovery tasks in an image deconvolution context. Over a second phase, *HOGMep* is evaluated through a biological application for a classification purpose.

7.3.1 Joint multi-spectral image segmentation and deconvolution

Experiments are performed on both synthetic ('Synth') and benchmark ('Peppers') color images (with $B = 3$ color channels) of size 64×64 and 128×128, respectively. They are represented in Figure 7.5 and Figure 7.8. Note that the number of classes is $L = 4$ for the 'Synth' image while — based on a visual inspection — L is expected to be equal to 6 for the 'Peppers' images.

Each channel from the original image is convolved by a blur operator and further corrupted by a Gaussian noise. Figure 7.4 illustrates the spectra of the three uniform (or 2D boxcar) point spread functions of different scale. All simulations was performed with a shape parameter β equal to 0.5. In these simulations, we restrict our HOGM to a Potts model and chose the Potts parameter providing the best results.



(a) (b) (c)

Figure 7.4 ~ SPECTRA OF THE UNIFORM BLUR OPERATORS ~
They are respectively applied on (a) Red, (b) Green and (c) Blue channels of the original color image, with a uniform boxcar point spread function of size $n \times n$ (with pixel value: $1/n^2$) with $n = 3, 5,$ and 7.

~ Restoration results ~ We assess the restoration performance, in terms of SNR, obtained with our approach *HOGMep*, VB-MIG (Ayasso and Mohammad-Djafari, 2010) — in which our more general MEP prior is restricted to a Gaussian one — and the recent state-of-the-art variational approach 3MG (Chouzenoux *et al.*, 2013). Note that not all approaches deal with multi-component images. We overcome this issue by comparing each channel in addition to the resulting overall reconstruction.

Reconstructed images for the 'Synth' image with various noise levels ($\sigma = 0.01, 0.05$ or 0.1) are displayed in Figures 7.5 to 7.6. The corresponding SNR results are provided in Table 7.1. Image reconstruction for 'Peppers' and various noise levels ($\sigma = 0.01$ or 1) are displayed in Figures 7.8 to 7.9. The associated SNR are gathered in Table 7.2. As expected, for all methods, SNR gains diminish with the (low-pass) bandwidth of the 2D boxcar blur operator and the noise level. Nevertheless, the proposed approach *HOGMep* leads, in the majority of cases, to higher objective SNR measures. Additionally, we can observe on the 'Synth' and the less noisy 'Peppers' images that, unlike for 3MG for which a slight blur remains, details in images (e.g. drapery, onion, etc. in 'Peppers') are well restored by the other methods. Using VB-MIG or *HOGMep*, some recovered pixels are sometimes wrongly colored according to unexpected segmentation results. Nevertheless, undesired over-pixelated regions are less present with *HOGMep* due to its better segmentation results (more detailed segmentation results in the following). Regarding the same 'Peppers' image with an increased noise (Figure 7.9), while debluring is correctly performed by the three tested algorithms, spurious noise artifacts remain. This effect in VB-MIG and *HOGMep* reconstruction is mainly due to segmentation results. Note that for 3MG, a simplex algorithm

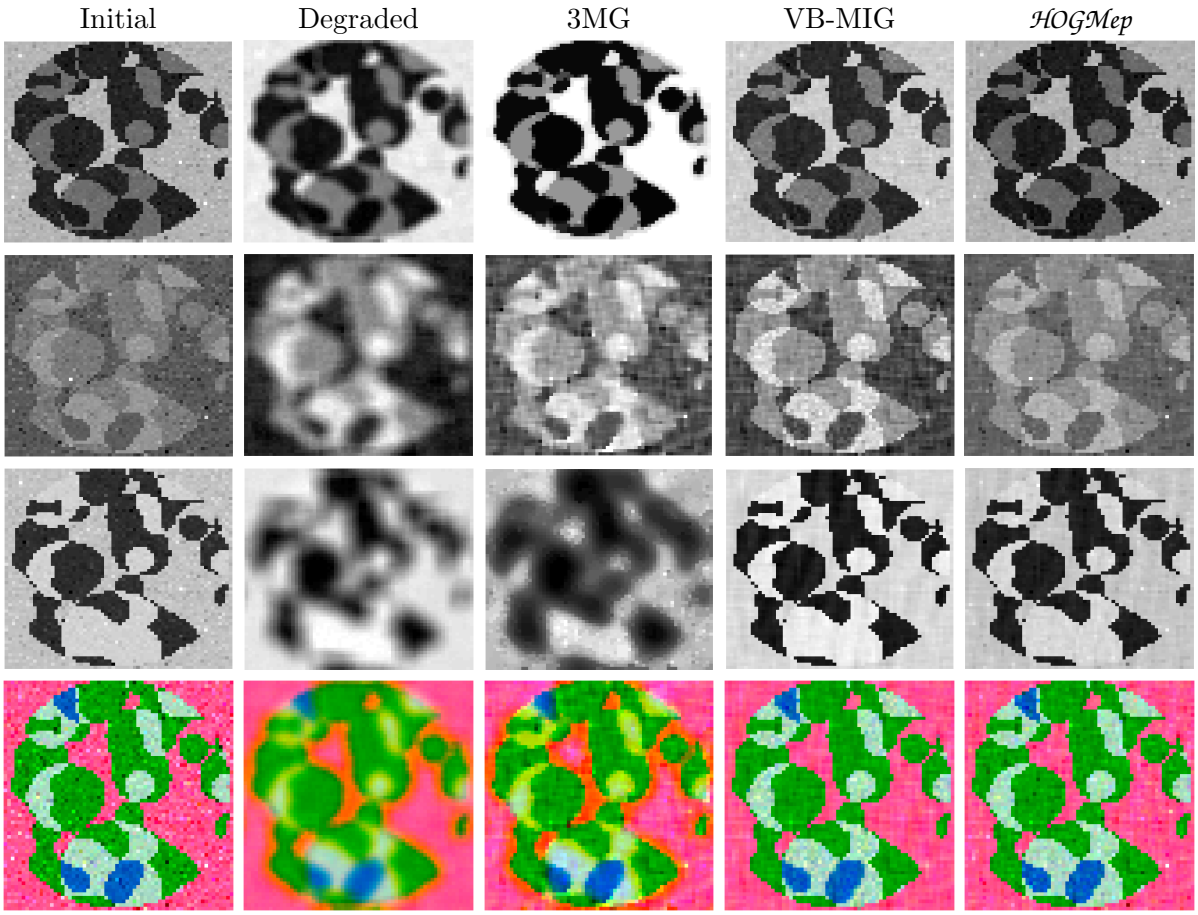


Figure 7.5 ~ Restoration results with noise level set to $\sigma = 0.01$ ('Synth') ~ *From top to bottom, R, G and B channels as well as the color image for the original, degraded and restored data with 3MG, VB-MIG and HOGMep.*

is used to determine an appropriate set of parameters. However, actual optimum not ensured, and observed spurious noise artifacts could be weakened by alternative parameter choices.

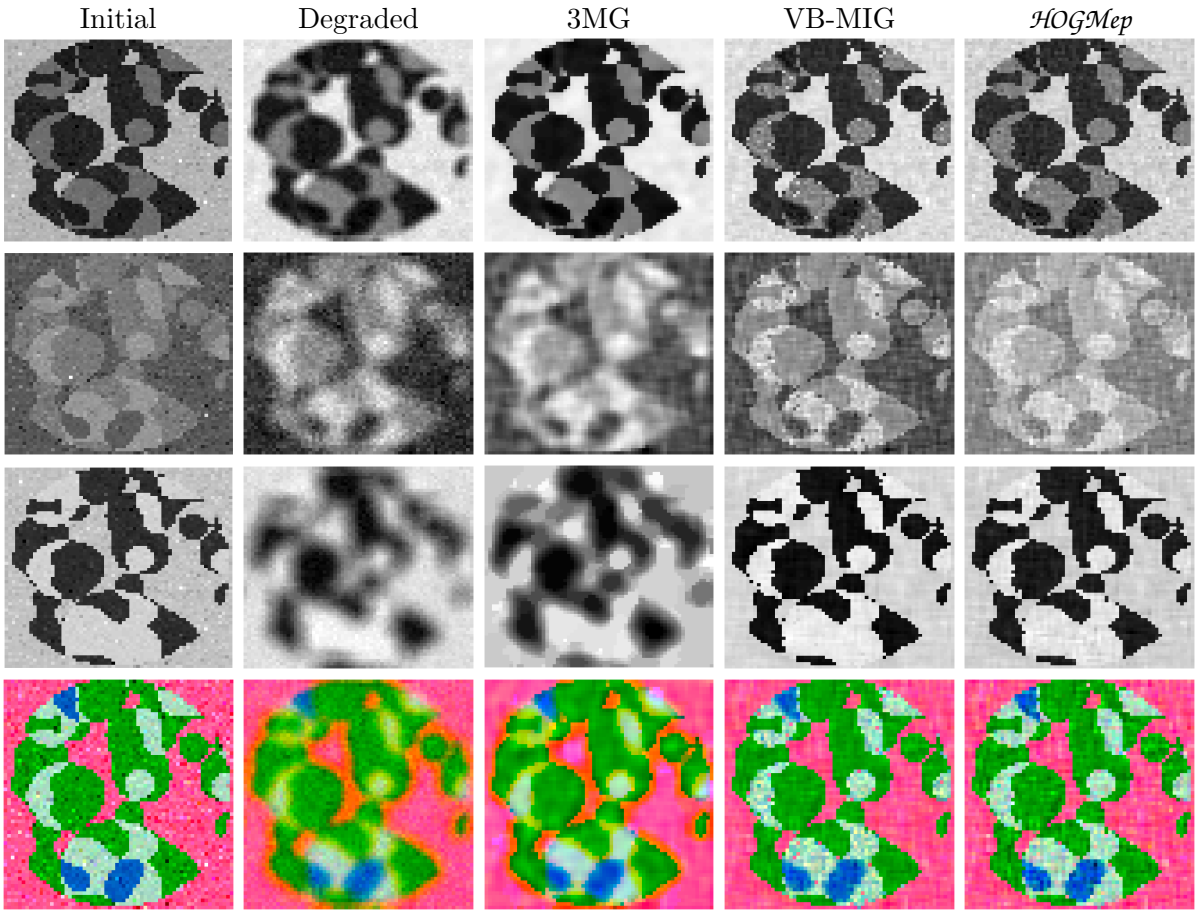


Figure 7.6 ~ Restoration results with noise level set to $\sigma = 0.05$ ('Synth') ~ *From top to bottom, R, G and B channels as well as the color image for the original, degraded and restored data with 3MG, VB-MIG and HOGMep.*

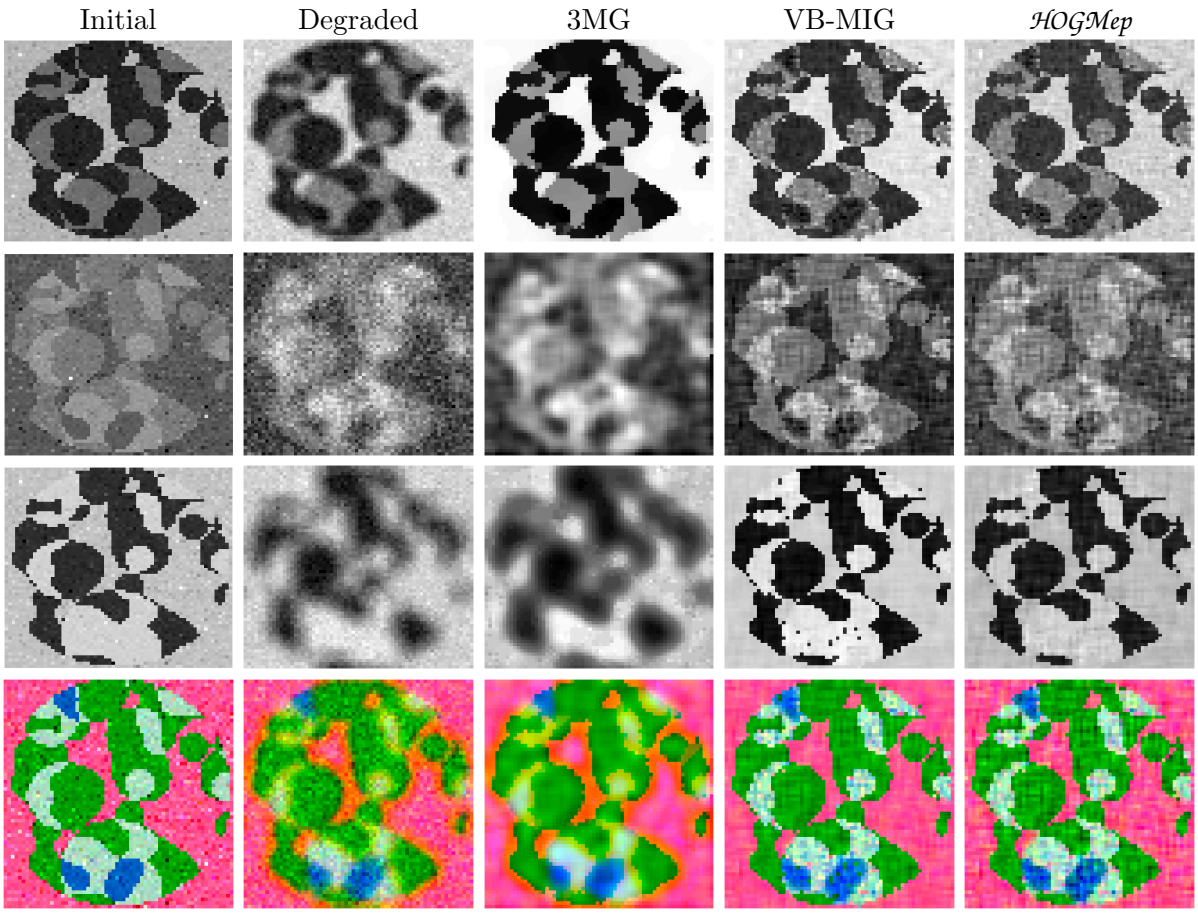


Figure 7.7 ~ Restoration results with noise level set to $\sigma = 0.1$ ('Synth') ~ *From top to bottom, R, G and B channels as well as the color image for the original, degraded and restored data with 3MG, VB-MIG and HOGMep.*

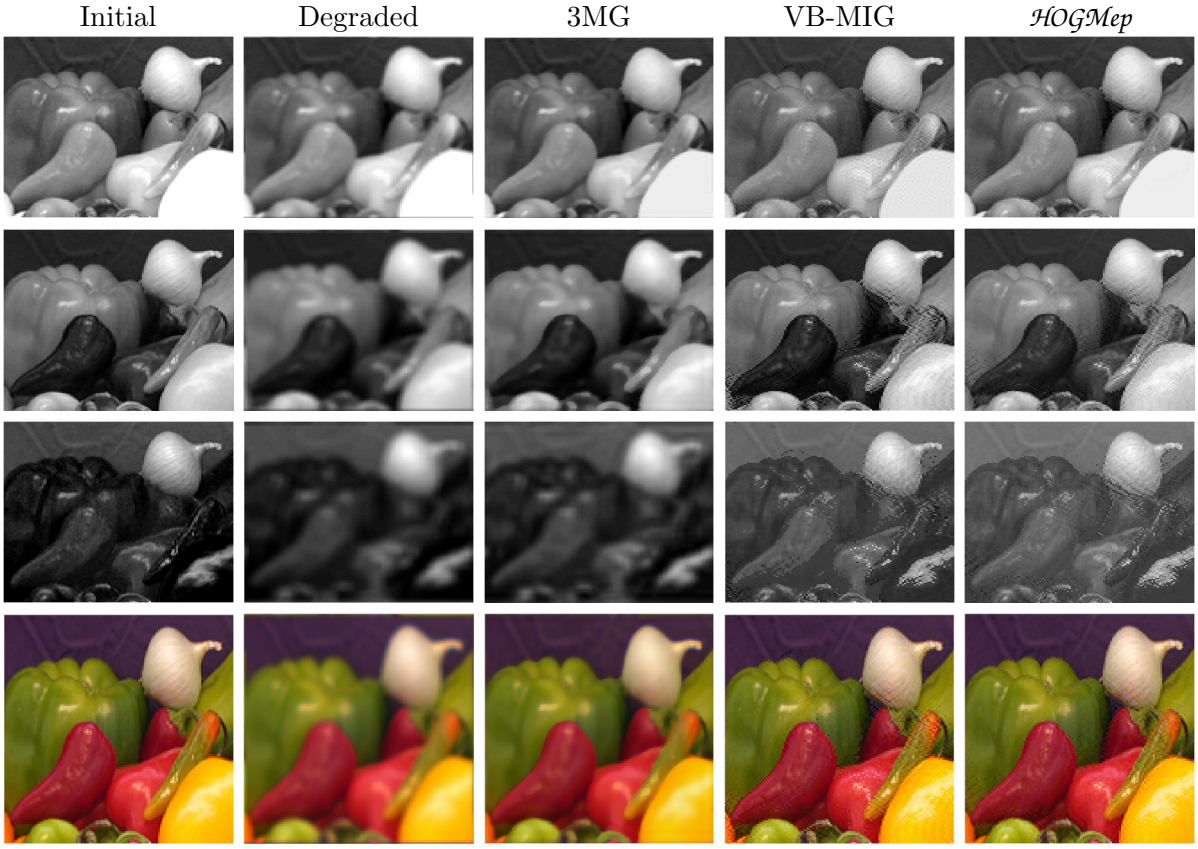


Figure 7.8 ~ Restoration results with noise level set to $\sigma = 0.01$ ('Peppers') ~ *From top to bottom, R, G and B channels as well as the color image for the original, degraded and restored data with 3MG, VB-MIG and HOGMep.*

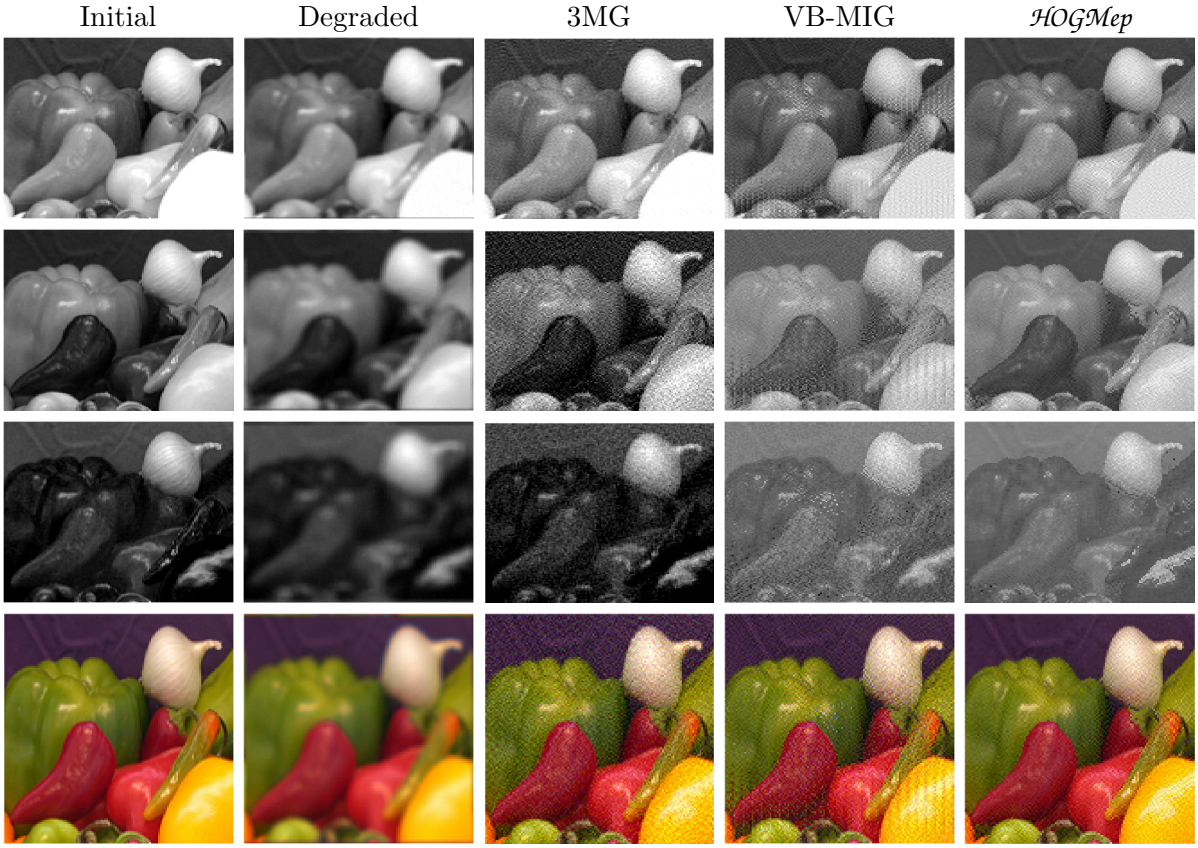


Figure 7.9 ~ Restoration results with noise level set to $\sigma = 1$ ('Peppers') ~ *From top to bottom, R, G and B channels as well as the color image for the original, degraded and restored data with 3MG, VB-MIG and HOGMep.*

Noise level	Method	Red	Green	Blue	Color
σ = 0.01	Initial	10.73	12.42	3.92	6.98
	3MG	11.21	15.17	2.60	6.08
	VB-MIG	22.62	16.41	18.39	19.41
	HOGMep	*24.25*	*17.14*	*19.75*	*20.63*
σ = 0.05	Initial	10.61	11.96	3.90	6.92
	3MG	16.33	13.16	4.73	8.41
	VB-MIG	16.71	*13.52*	12.96	14.47
	HOGMep	*17.68*	13.28	*13.87*	*15.15*
σ = 0.1	Initial	10.28	10.44	3.75	6.65
	3MG	12.29	12.46	3.30	6.74
	VB-MIG	*14.90*	*12.82*	10.96	12.74
	HOGMep	14.85	11.80	*11.61*	*12.90*

Table 7.1 ~ Channel and color restoration results in terms of SNR ('Synth'). ~
Best performers are in italics. 'Initial' rows refer to the initial image, without any processing.

Noise level	Method	Red	Green	Blue	Color
σ = 0.01	Initial	24.30	18.36	13.77	19.70
	3MG	28.99	22.82	14.19	22.25
	VB-MIG	29.94	21.13	14.12	21.77
	HOGMep	*33.43*	*23.96*	*14.30*	*23.00*
σ = 1	Initial	24.24	18.33	13.73	19.65
	3MG	*26.16*	13.69	*13.50*	17.19
	VB-MIG	20.53	14.05	9.45	15.49
	HOGMep	22.80	*19.69*	13.08	*19.56*

Table 7.2 ~ Channel and color restoration results in terms of SNR ('Peppers'). ~
Best performers are in italics. 'Initial' rows refer to the initial image, without any processing.

~ *Segmentation results* ~ Segmentation results obtained by *HOGMep* are compared to those obtained with VB-MIG (Ayasso and Mohammad-Djafari, 2010), the iterated conditional mode (ICM) algorithm (Besag, 1986) and the spectral clustering (SC) algorithm (Ng *et al.*, 2001). Note that not all segmentation methods integrate a degradation modeling in their formulation. Hence, segmentation performance can thus be evaluated from degraded images and non-degraded images. For the latter, we chose to use the restored images provided by *HOGMep*. In such a case, we may demonstrate the interest to integrate degradation modeling for segmentation enhancement. Two strategies are employed to evaluate the clustering according to the availability of a ground truth or not. For the 'Synth' image, a ground truth is available. We thus numerically assess segmentation results through Variation of Information (VI) (Meilă, 2003) and Rand Index measures (RI) (Rand, 1971). We recall that the description of these measures is provided in Section 3.2.3. These measures are provided in Table 7.3 while segmented images and their binary difference to the original segmentation are displayed in Figures 7.10 to 7.12. As it can be observed through the experiments, *HOGMep* offers a very low rate of wrongly assigned pixels (around 2 % for the maximum), and globally correctly labels all regions. While good performances are obtained using a spectral clustering algorithm from the restored image, they decrease when the initial image is degraded. This observation is in favor of our joint procedure, where the segmentation benefits from degradation modeling.

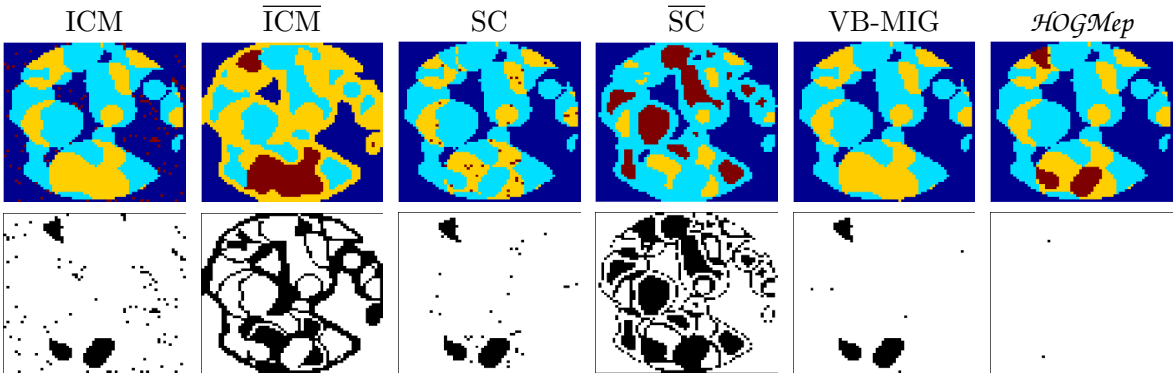


Figure 7.10 **~** SEGMENTATION RESULTS WITH NOISE LEVEL SET TO $\sigma = 0.01$ ('SYNTH') **~** *Segmentation results obtained using Iterated Conditional Mode on restored (ICM) and degraded image ($\overline{\text{ICM}}$), Spectral Clustering on restored (SC) and degraded image ($\overline{\text{SC}}$), VB-MIG and HOGMep. The binary difference to original ground truth is also provided with wrong pixels in black.*

Now, regarding the 'Peppers' image, for which no ground truth for the segmentation is available, another assessment strategy is employed. It is based on a intrinsic evaluation of the segmentation thanks to the silhouette measure (Rousseeuw, 1987) — for which details are given in Section 3.2.3. We recall that this measure estimates the clustering performance intrinsically by assigning a silhouette value between −1 and 1 at each pixel. A silhouette value close to 1 indicates that the pixel cannot be better associated to another cluster while a silhouette tending to −1 is obtained when the pixel would be better associated to another cluster — its nearest

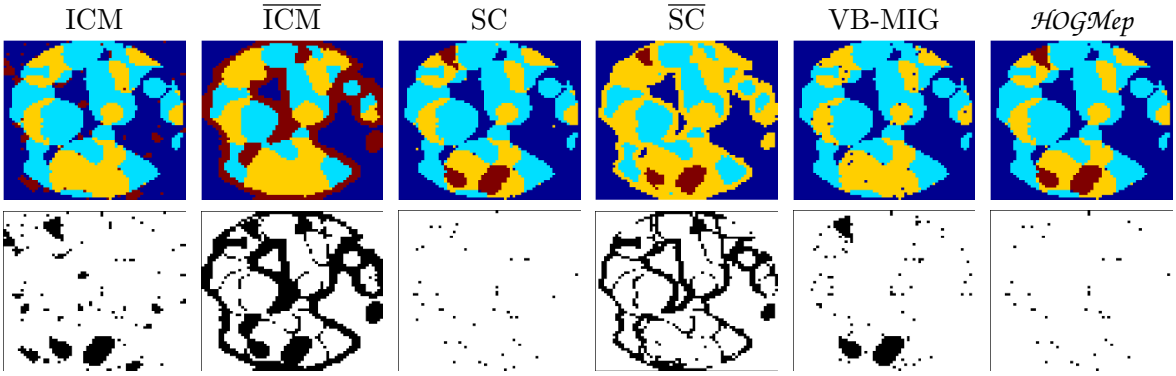


Figure 7.11 ~ SEGMENTATION RESULTS WITH NOISE LEVEL SET TO $\sigma = 0.05$ ('SYNTH') ~ *Segmentation results obtained using Iterated Conditional Mode on restored (ICM) and degraded image ($\overline{\text{ICM}}$), Spectral Clustering on restored (SC) and degraded image ($\overline{\text{SC}}$), VB-MIG and HOGMep. The binary difference to original ground truth is also provided with wrong pixels in black.*

in the sense of a similarity criterion. Finally, when the silhouette is equal to 0, the pixel is at the border of the two clusters and the interpretation regarding the best cluster is then difficult. At a global image scale, it is common to study the average or median silhouette measures. In addition, we can consider that a silhouette measure higher than 0.5 reflects a significant assignment to the most plausible cluster. We thus choose to assess the proportion of pixels having a silhouette score higher than 0.5. Results using this evaluation methodology are provided in Table 7.4. Segmented images are displayed in Figures 7.13 to 7.14.

Regarding the average and median silhouette, *HOGMep* is the second-best segmentation method (the first one differing in each case). However, when we compare the number of significantly well-assigned labels, *HOGMep* leads to better segmentation results. It appears that it can be difficult to provide a rigorous conclusion on the silhouette-based tested criterion. We overcome this by assessing clustering performance through the image of the silhouette measures, see Figures 7.13 to 7.14. In such images, blackish pixels reflect silhouette tending to -1 while close-to-white pixels correspond to a silhouette score close to 1. Hence, white and light gray pixels are preferred to the others. Based on this criterion, *HOGMep* leads to a better segmentation results than those obtained with the ICM, SC or VB-MIG methods. This conclusion can also be drawn from the histogram of silhouette measures, see Figures 7.15 to 7.16. Notably for the less noisy image, ICM and *HOGMep* present the higher number of well-classified pixels (silhouette tending to 1) over all the tested methods. However, ICM also exhibits a non-negligible number of misclassified (silhouette index close to -1) pixels suggesting *HOGMep* to be the overall best performer. Note that SC presents intermediate results while VB-MIG is the method having the lower number of well-classified (silhouette close to 1) pixels. Results are similar for higher noise levels: *HOGMep* shows the best compromise between a high level of silhouette close to 1, and a low level close to -1. Note that, as VB-MIG only segments the image into two classes instead of 6, comparing silhouette measures can appear irrelevant.

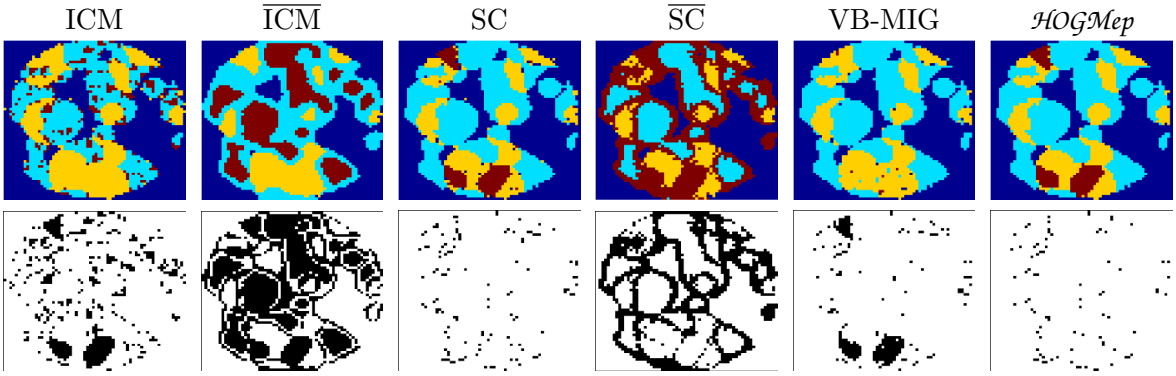


Figure 7.12 ~ SEGMENTATION RESULTS WITH NOISE LEVEL SET TO $\sigma = 0.1$ ('SYNTH') ~ *Segmentation results obtained using Iterated Conditional Mode on restored (ICM) and degraded image ($\overline{\text{ICM}}$), Spectral Clustering on restored (SC) and degraded image ($\overline{\text{SC}}$), VB-MIG and HOGMep. The binary difference to original ground truth is also provided with wrong pixels in black.*

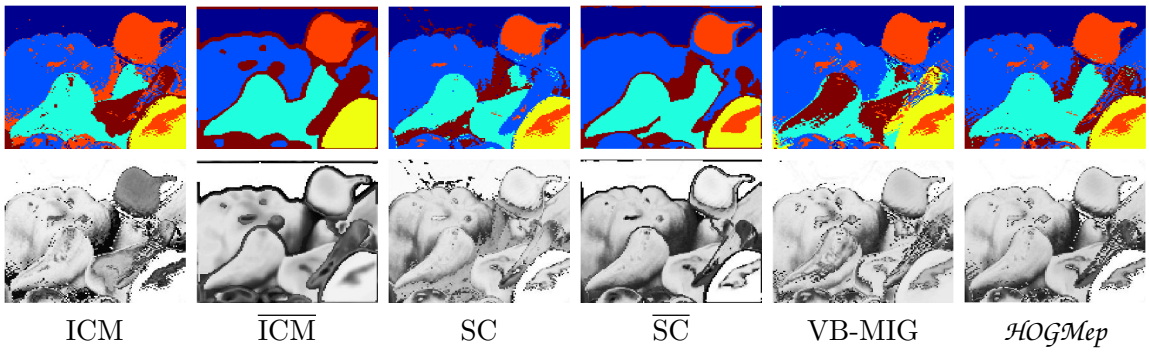


Figure 7.13 ~ SEGMENTATION RESULTS WITH NOISE LEVEL SET TO $\sigma = 0.01$ ('PEPPERS') ~ *Top row: one color encodes one class. Bottom row: silhouette image in which black-trend pixels reflect silhouette tending to −1 while white-trend pixels correspond to a silhouette score of 1. Using this color code, white-trend zone reflects satisfying clustering.*

7.3.2 Biological application

In the medical field, it is common to classify a population of patients into two classes: healthy or ill, for instance. This classification may be based on various types of data, including gene expression levels. To illustrate this kind of biological application, let us briefly describe the benchmark 'breast cancer' dataset (Hess *et al.*, 2006). It corresponds to the expression level of 26 genes for 133 patients with stages I-III cancer. Patients were treated with chemotherapy before having to undergo a surgical procedure. The patient response to the treatment is thus classified into two classes: pathological complete response (pCR) — 34 patients — or residual disease (not-pCR) — 99 patients.

Noise level		ICM	$\overline{\text{ICM}}$	SC	$\overline{\text{SC}}$	VB-MIG	*HOGMep*
$\sigma = 0.01$	VI	0.209	1.192	0.177	1.050	0.130	*0.006*
	RI	92.92	65.04	96.51	76.44	98.24	*99.05*
	% misclass.	7.08	34.92	5.35	34.42	4.66	*0.05*
$\sigma = 0.05$	VI	0.336	1.122	0.119	0.956	0.284	*0.109*
	RI	90.72	64.96	98.96	77.28	93.72	*99.05*
	% misclass.	9.28	35.03	1.05	23.51	6.27	*0.95*
$\sigma = 0.1$	VI	0.471	1.122	0.274	1.073	0.341	*0.208*
	RI	86.64	64.87	97.16	78.24	93.53	*97.95*
	% misclass.	13.35	44.12	2.86	31.86	6.47	*2.05*

Table 7.3 ~ SEGMENTATION RESULTS IN TERMS OF VI ('SYNTH') ~
Best performers are in italics.

		ICM	$\overline{\text{ICM}}$	SC	$\overline{\text{SC}}$	VB-MIG	*HOGMep*
$\sigma = 0.01$	S_{mean}	0.4506	0.3195	*0.5514*	0.4194	0.4842	0.4582
	S_{med}	*0.6924*	0.5807	0.6902	0.6375	0.6510	0.6525
	S (%)	62.0	55.4	64.8	61.9	66.0	*66.4*
$\sigma = 1$	S_{mean}	0.4783	0.3347	0.2583	0.3370	*0.6313*	0.3851
	S_{med}	0.5626	0.5351	0.5272	0.4865	0.4443	*0.5658*
	S (%)	55.9	52.9	51.7	48.6	44.2	*57.2*

Table 7.4 ~ SEGMENTATION RESULTS IN TERMS OF SILHOUETTE ('PEPPERS') ~
Quantities S_{mean} and S_{med} denote the average and median silhouette measures over all pixels, respectively. **S** *refers the proportion of pixels having a silhouette score higher than 0.5 (significant assignment to the most plausible cluster). Best performances are highlighted in italics.*

To answer the aforementioned unsupervised classification problem, the *HOGMep* model can be adapted. Using the aforementioned 'breast cancer' dataset, we have at our disposal $N = 133$ variables for which $B = 26$ observations (gene expression levels) are available. As the problem is restricted to classification only, the degradation operator $\mathbf{H}$ is the identity matrix. Note that, in such kind of application, the degradation operator could encode smoothing to reduce clinical variability. Based on these gene expression levels, patient classification can be performed by setting the number or desired classes L to 2 (pCR or not-pCR).

In this biological context, *HOGMep* performance (restricted to a Potts model and a MEP shape parameter β set to 0.5) is compared to those obtained with either a K-means or a spectral clustering algorithm. Results in terms of variation of information (VI) (Meilă, 2003) and Rand index (RI) (Rand, 1971) are provided in Table 7.5. As highlighted in italics, the best objective classification is recovered using *HOGMep*. Illustration of the resulting clusterings are

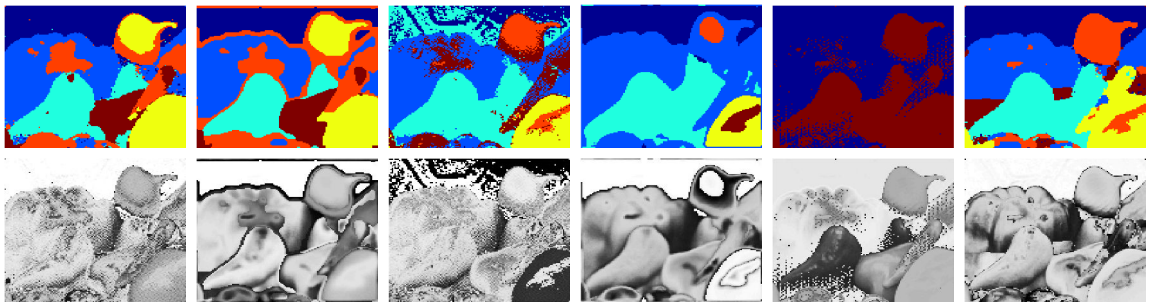


Figure 7.14 ~ Segmentation results with noise level set to $\sigma = 1$ ('Peppers') ~ *Top row: one color encodes one class. Bottom row: silhouette image in which black-trend pixels reflects silhouette tending to −1 while white-trend pixels correspond to a silhouette score of 1. Using this color code, white-trend zone reflects satisfying clustering.*

displayed in Figure 7.17.

	SC	K-means	*HOGMep*
VI	0.8812	0.8412	*0.8267*
RI	77.44	79.70	*80.45*

Table 7.5 ~ Numerical performance for breast cancer data classification ~

Note that using *HOGMep* with a shape parameter β of the MEP distribution set to 1 i.e. Gaussian prior, classification results are weaker. Indeed, the obtained VI and RI reach 0.8811 and 77.44, leading to *HOGMep* gains reaching about 6 % and 4 %, respectively. This result demonstrates the potential interest of using a MEP prior instead to restricting it to a Gaussian one.

7.4 Conclusions on *HOGMep*

Inference and clustering have so far been considered in *BRANE Cut*, *BRANE Relax* and *BRANE Clust* in the framework of gene networks. They can be embedded into a larger scope of reconstruction, classification or segmentation for a wider class of graph-structured data representations. Indeed, some of the tools we used have demonstrated their effectiveness on images, which can be considered as graphs with specific connexity. Additionally, the variational approach in *BRANE Clust* bears analogy with Potts models (Section 6.2.1). The latter can be plasticized into a Bayesian framework to broader distribution families. Notably, in *HOGMep* — which models inverse problems through a Bayesian framework for joint segmentation and recovery — a Higher Order Graphical Model (HOGM) is employed on latent label variables for clustering or classification. In addition, a Multivariate Exponential Power (MEP) prior is opted for the signal in a given class. An efficient Variational Bayesian Approximation (VBA) was developed to solve the as-

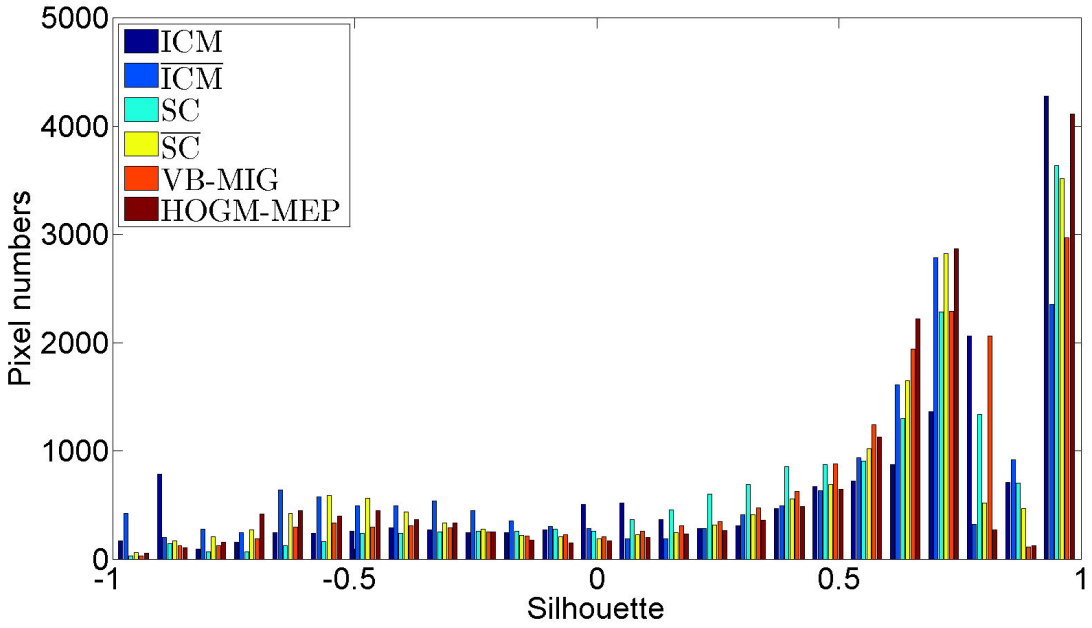


Figure 7.15 ~ HISTOGRAM OF SILHOUETTE FOR NOISE VARIANCE $\sigma = 0.01$ ('PEPPERS') ~ *Silhouette measures computed at each pixel for ICM, spectral clustering (from restored (SC) and degraded images ($\overline{\text{SC}}$)) VB-MIG and HOGMep.*

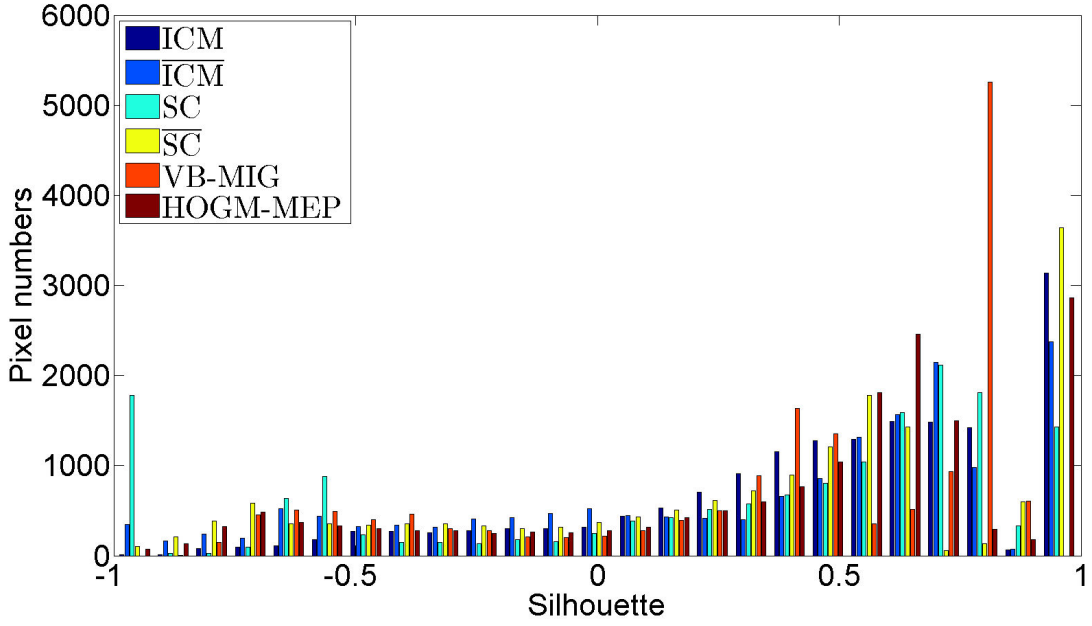
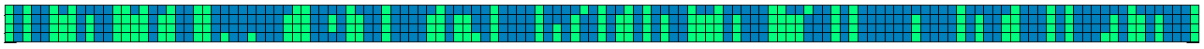


Figure 7.16 ~ HISTOGRAM OF SILHOUETTE FOR NOISE VARIANCE $\sigma = 1$ ('PEPPERS') ~ *Silhouette measures computed at each pixel for ICM, spectral clustering (from restored (SC) and degraded images ($\overline{\text{SC}}$)) VB-MIG and HOGMep.*



(a) Final classification.

(b) Difference to the ground truth.

Figure 7.17 ~ SEGMENTATION RESULTS FOR THE BREAST CANCER DATASET ~
(a) Colored pixels refer to the class: green for pCR and blue for not-pCR. Classified variables are given in abscissa and each row corresponds to one classification methods. From top to bottom: ground truth, HOGMep, K-means and spectral clustering. (b) Difference to the ground truth: badly classified variables are represented as black pixels.

sociated problem. Its flexibility accommodates a broad range of applications, demonstrated on multi-compoment image deconvolution coupled to segmentation and an initiatory venture into medical data classification.

8 Conclusions and perspectives

"Above all, don't fear difficult moments. The best comes from them"

Rita Levi-Montalcini

8.1 Conclusions

In biology, treatment of "omics" data requires cross-disciplinary skills, at the intersection of biotechnologies, computer science and mathematics. In this context, this thesis focused on the development of bioinformatics tools allowing the inference of Gene Regulatory Networks (GRNs) from transcriptomic data. GRNs are graphs for which nodes and edges are respectively derived from genes and their correlations or regulations. Their construction is useful for sketching transcriptional regulatory pathways and helping to understand phenotype variations.

8.1.1 *BRANE* strategy: gene networks as graphs and a priori-based optimization

Modeling transcriptomic data through graph structures allows us to take advantage of known and efficient graph-based algorithms for directly providing GRNs. Indeed, we propose to address network inference problems with recently developed techniques pertaining to graph (nodes and edges) optimization with biological and structural priors, leading to a variational analysis formulation. Those choices are driven by an apparent simplicity of certain basic biological mechanisms in cells[1], and the ability to operate with conditions allowing variations in the studied enzyme production mechanisms.

From a complete weighted gene network, where weights computed from transcriptomic data encode the strength of the interaction between genes, our proposed *BRANE* approaches aim at selecting subsets, ideally reflecting gene regulatory links. This analysis unveils experimental venues for novel interactions and gene function identification. For this purpose, binary variables encoding the edge presence or absence in the final graph have been defined. From them, our strategy is to firstly reformulate the classical edge selection — assimilated to a simple thresholding — into an optimization problem. This basic formulation as a cost function was then

[1] However, after Heinrich Hertz, *we cannot a priori demand from nature simplicity, nor can we judge what in her opinion is simple.*

improved by the integration of regularization terms encoding biological and structural *a priori*. Our priors are defined from the knowledge, often available, of particular genes coding for transcription factors — proteins which regulate gene expression. Different priors lead to different mathematical properties of the global cost function, for which various optimization strategies can be applied. Based on this strategy, the post-processing network refinements we proposed in this thesis led to a set of computational approaches named *BRANE* for "Biologically-Related A priori for Network Enhancement".

BRANE Cut This first edge selection strategy is designed to favor the selection of strongly weighted edges as in classical thresholding. Additionally, modular networks arranged around transcription factors are promoted by defining a threshold parameter with respect to the nature of genes. A regularization term, based on an edge inference coupling and enforcing gene co-regulation is also added. The latter is weighted by a probability of co-regulation, computed from the gene expression data. The resulting cost function is an instance of a minimum cut energy function. Thanks to the minimum cut/maximal flow duality, our discrete optimization problem is solved thanks to a maximal flow algorithm. For this purpose, an intermediate transportation network is constructed with respect to the underlying cost function. Our proposed approach is validated on both simulated datasets provided in the DREAM4 and DREAM5 challenges and real datasets from the bacteria *Escherichia coli* and the fungus *Trichoderma reesei*. Our simulations show significant improvement — in terms of numerical performance and biological interpretation of inferred networks — over state-of-the-art methods (CLR, GENIE3 and the Network Deconvolution (ND) post-processing).

BRANE Relax While strongly weighted edges and network modularity are enforced in a similar way to *BRANE Cut*, this edge selection strategy differs from the previous one by the integration of a connectivity *a priori* instead of a gene co-regulation ones. This connectivity *a priori*, defined for genes not identified as coding for transcription factors, allows us to restrict their node degree, according to biological knowledge. To overcome the fact that the resulting discrete optimization problem cannot be efficiently solved using discrete optimization, a continuous relaxation is employed. A proximal splitting scheme is then applied. As a result, the relaxed optimization problem is solved using a projected gradient algorithm. Convergence speed of the latter is improved by the combination of two recent tricks: a preconditioning using the Majorize-Minimize principle and a block coordinate strategy. The resulting approach is validated on the simulated datasets from the DREAM4 and DREAM5 challenges. Comparisons with state-of-the-art methods CLR, GENIE3 and ND demonstrate significant improvements in terms of numerical performance.

BRANE Clust The last edge selection strategy we proposed differs from *BRANE Cut* and *BRANE Relax* as it was devised to integrate gene clustering — a task traditionally performed in an independent manner in the classical workflow of transcriptomic data analysis. The gene clustering is used to penalize the selection of edges linking nodes assigned to a distinct cluster. Additionally, the gene clustering was constructed to generate modules centered

around transcription factors in order to promote modular networks. As a results, in addition to edge label variables, node label variables (encoding the cluster label assignment) have to be determined. This problem is thus solved through an alternating optimization strategy. While the optimal edge labels are explicitly obtained, optimal node labels are obtained using a random walker algorithm after relaxing the underlying sub-problem. This approach is validated on both simulated and real datasets provided by the DREAM4 and the DREAM5 challenges. Comparisons to state-of-the-art methods CLR, GENIE3 and ND are in favor of our approach with significant improvements in terms of numerical performance and, more importantly, of biological interpretation of inferred networks.

8.1.2 *HOGMep* for a wide graph-based processing

A ramification of this thesis is focused on more generic graph-based problems. While inference consists in the structure recovery, inverse problems can be considered in this context since signal and images can be easily modeled through a graph structure. In this respect, a preliminary work relies the evaluation of the capabilities of *HOGMep*, a Bayesian approach we developed for joint reconstruction and clustering on multi-component data. For this purpose, two kinds of validations were performed according to the nature of the data: structured or unstructured. We thus firstly demonstrate the performance of *HOGMep* in an image deconvolution context. Simulations on synthetic and benchmark color images show significant improvements over its competitors in terms of pixel recovery (reconstruction) and pixel classification (segmentation). Promising preliminary results for genotype/phenotype medical data classification have also been obtained.

8.2 Perspectives

As regularly mentioned over this manuscript, GRN inference, even combined with clustering, is only one tool for transcriptomic data treatments — traditionally mutually independent and based on various, distinct and sometimes unconnected hypotheses. While the *BRANE* strategy we developed in this thesis provides a solid start-point for GRN inference enhancement, it could be interesting to work on the preceding treatments, notably the normalization of transcriptomic data and the detection of differentially expressed genes.

8.2.1 Biological-related perspectives

~ Normalizing transcriptomic data with biology-related a priori ~ As introduced in Section 2.3.1, a large variety of normalization procedures — with their own assumptions — exists for both DNA microarray and RNA-seq data. While the lowess normalization is admitted to well perform on DNA microarray data, no consensus emerges for RNA-seq data (Dillies *et al.*, 2013; Evans *et al.*, 2017) and we are seeing renewed interest in RNA-seq normalization (Hicks and Irizarry, 2015). In addition, a new trend, consisting in DNA microarray and RNA-seq aggregation, emerges (Taroni and Greene, 2017). As a first perspective, we would like to develop a normalization procedure which could be applied, in a less distinct and *ad-hoc* manner, on both a given microarray and RNA-seq data experiment, for both biological replicates and

inter-condition normalization. This ambition raises several challenges due to the heterogeneity in expression and scale, and outliers in the data (Gierliński *et al.*, 2015). Nevertheless, a possible trail emerged for RNA-seq data which could be subsequently adapted for DNA microarray data. We recall that RNA-seq experiments, in a given experimental condition, provide for each gene a read count. Restricting here experimental conditions to biological replicates, we expect that for subsets of genes, gathered from *a priori* or inference, read counts over biological replicates are relatively similar. In such a case, we expect more a robust correction of experimental biases and dispersions from more homogeneous groups of genes. This task involves recent developments on optimization with respect to robust regression, sparsity and outlier weighting. For this purpose, genes having a relatively coherent behavior can be firstly detected — in an automated manner — in order to use them as reference genes for normalization factor determination. This reasoning could also be adapted for a normalization across the experimental conditions.

~ *Restoring and identifying DE genes from microarrays with HOGMep* **~** As demonstrated through our simulations, *HOGMep* is a suitable tool for performing joint high-quality reconstruction and classification of multi-component data. An elegant application could be investigated in the context of transcriptomic data. We recall that microarray experiments firstly provide images containing colored spot. From these images, red and green intensities are computed for each spot, yielding the gene expression data we know. These intensities are then normalized and used to detect differentially expressed genes. A second perspective is to use *HOGMep* to provide the set of differentially expressed genes from the microarray images directly. For a given set of experiments, resulting in a set of microarray images and assuming that spots are coherent from a microarray to another i.e. each spot corresponds to the same gene in each microarray, this set of images could be used as multi-component data that can be processed by *HOGMep*. One challenge is to properly defined the degradation operators involved in the model. Indeed, they should be defined so that pixel backgrounds in each spot are removed additionally encoding some normalization aspects. As a result, we expect that, in the same time of the recovery of true intensities, *HOGMep* also provides a classification of the spots by discriminating those for which the intensities are relatively constant across the set of microarray images. Such spots could thus be assimilated to non differentially expressed genes.

~ *Finding shortcuts in transcriptomic data processing workflows* **~** Dealing with transcriptomic data requires to master and to parametrize a complicated pipeline of analyses from gene expression level measurement to useful information extraction. Such pipelines involve for instance normalization, differential analysis, correlation, clustering and network inference. In addition to be specific to the studied data (DNA microarray or RNA-seq), often platform-dependent, the different steps of pipelines involve various distinct assumptions, which can be prejudicial to unconventional studies. Based on this statement, one challenging task is to elaborate a better integrated pipeline, regarding the treatments as much as the data. In such a pipeline, not all treatments need to be coupled but they have to make a minimum of (hopefully biologically-related) assumptions. For instance, a suitable way of reducing the pipeline depth could probably lie in the merging of normalization and differential analysis on the one hand, and the merging of clustering and network inference (in the manner of *BRANE Clust*) on the

other hand. Notably, the latter part could be directly obtained through gene expression profiles themselves instead of losing information by integrating a weight computation step, especially sensitive on very short data runs. In addition to more integrative and flexible treatments, the integration of the data themselves could be considered. As it is now common to have both DNA microarray and RNA-seq data at our disposal, it could be judicious to work on a data-integrated workflow to take advantage of the two complementary technologies instead of performing independent analyses, for which the cross-check of the results can appear challenging. It is obvious that this data integration challenge also gives rise to normalization open problems.

~ *Integrating larger scale regulation processes* **~** While the work provided in this thesis was focused on the construction of GRNs from transcriptomic data only, it could also be interesting to investigate other -omics scales. Indeed, in addition to purely genetic regulation *via* transcription factors, other gene regulatory mechanisms occur at various scales such as chromatin condensation, promoter methylation or through miRNA. Such processes take part of the fast-growing field of the epigenetic — biological phenomena impacting the gene regulation but not encoded in the genome itself. Related information can be obtained with specific complementary experiments such as HiC, single-cell RNAseq or ChipSeq to name a few, and for which specific and independent treatments are developed (Liang and Keleş, 2012; Lévy-Leduc *et al.*, 2014; Quang and Xie, 2014; Kharchenko *et al.*, 2014; Servant *et al.*, 2015). From an optimal viewpoint, integrating these various regulatory schemes could be very informative for the discovery of gene regulatory pathways. Their processing may require other sets of data processing tools, such as the ones we lightly touched on baseline separation for analytical signals. However, the integration and the treatments of these highly heterogeneous data are very challenging but should be seriously considered in future works. Gene regulatory processes are complex and integrated. Their discovery thus requires a higher level of preservation of the richness and complementarity present in data under their rawest form.

8.2.2 Signal/image-related perspectives

~ *HOGMep extensions to blind inverse problems and non-Gaussian noise* **~** In this thesis, we present *HOGMep*, a Bayesian approach designed to jointly perform restoration and classification on multi-component signals or images. We recall that a higher-order graphical model is used for the prior distribution on the latent variables encoding the cluster label assignment. Not demonstrated in our simulations, where only a special case is used i.e. the Potts model, the interest of such a generic modeling — in an image segmentation (Kohli *et al.*, 2009) or classification in social and affiliation networks (Zheleva *et al.*, 2010), for instance — is a natural first-order perspective.

In a farther perspective, while *HOGMep* solves non-blind inverse problems, it could be interesting to extend our approach to blind inverse problems, largely encountered in signal/image processing. In such a case, an additional prior probability has to be proposed to model the unknown degradation operator. Note that this prior has to be chosen in accordance with the characteristics of the degradation operator, often specific to a particular application such as blur deconvolution, thus restricting the field of application of *HOGMep*. According to the novel joint probability distribution, a suitable VBA-based strategy for its estimation could be devel-

oped. Finally, another conceivable work on *HOGMep* relies on the noise statistical assumption. In our work, the observation model we use assumes an additive Gaussian noise. Nevertheless, in some applications, it could be interesting to deal with other noise modeling: Poisson-Gaussian mixtures, Poisson or log-normal, for instance.

~ *HOGMep for arbitrary network inference* ~ In a network inference perspective, one could benefit from the segmentation capability of *HOGMep* on multi-component signal/image. Suppose that we have at our disposal a set of N edge-valued graphs $(\mathcal{G}_i)_{1\leq i\leq N}$, containing the same set of M nodes $\mathcal{V}$ and for which at each graph $\mathcal{G}_i$, an adjacency matrix $\mathbf{W}_i \in \mathbb{R}^{M\times M}$ is associated, reflecting the graph topology. Each adjacency matrix could encode a particular type of weights. In such a case, we can define a new graph $\mathcal{G}$ with $M \times M$ nodes, for which each node thus corresponds to an edge on the graph $\mathcal{G}_i$. This is called the dual graph. Hence, each node is valued by a vector of N weights from the adjacency matrices $(\mathbf{W}_i)_{1\leq i\leq N}$. The node-valued graph $\mathcal{G}$ can thus be used in *HOGMep* for network inference purposes. Indeed, the network inference process could thus result in a binary segmentation of the nodes in $\mathcal{G}$, driven by a variety of weights. In addition, if non-symmetric weights are available, the resulting inferred network could be directed. Indeed, this advantage is due to the duality used between edges and nodes and results in the fact that all edges are taken into account in generating the graph $\mathcal{G}$ to be segmented. In the same vein, this strategy could be adapted to data which can be described through multiple modalities yielding various graph structures from the same set of nodes (Dong *et al.*, 2012).

~ *Inference and graph topology constraint* ~ The *BRANE Relax* approach developed during this thesis allows us to infer modular networks by constraining the connectivity degree of some particular genes thanks to a regularization term incorporated in the optimization formulation of the classical thresholding. As a perspective of this work, the regularization term could encode a constraint on other kinds of topology. Notably, scale-free networks are encountered in a large number of applications such as social, computer, and financial networks to name a few (Clauset *et al.*, 2009). Such networks are characterized by a degree distribution following a power law. Based on this knowledge, the edge selection could be constrained such that the connectivity degree distribution of the inferred network is as close as possible to the theoretical distribution. Such a constraint can be encoded through a regularization term corresponding to a metric quantifying the difference between two distributions. For this purpose the divergence of Kullback-Leibler or the Hellinger distance could be used. In such a case, the challenge would be to find an appropriate optimization strategy to solve the related constrained problem.

~ *Laplacian-based tools for graph analysis* ~ Dealing with graphs can be performed through various structures including adjacency or incidence matrices and graph Laplacian. The latter is defined as the difference between the degree matrix (diagonal matrix in which diagonal element i corresponds to the degree of the node i) and the adjacency matrix. The graph Laplacian is commonly used for graph clustering (Grady and Polimeni, 2010; Van De Ville *et al.*, 2017) as in *BRANE Clust*. Due to the definition of the graph Laplacian, the latter can be used to infer the topology of a graph from observations (Dong *et al.*, 2016). Thanks to the ability of the

graph Laplacian to provide both graph topology and node clustering, an interesting perspective could reside in the development of a Laplacian-based joint inference and clustering approach. In addition to graph inference / reconstruction / clustering, another topic, not deeply investigated but simply used during this thesis, concerns the open problem of graph comparison. Indeed, one of the most used standard approaches is based on simple classifier metrics such as Precision and Recall. Unfortunately, they do not take into account the network topology but the edge presence/absence only. In order to refine graph comparison, it thus could be interesting to take advantage of the graph topology (Emmert-Streib *et al.*, 2012). The latter can notably be studied in the graph spectral domain (Shuman *et al.*, 2013). Indeed, the smoothness in the graph spectral domain provides information regarding the degree of connectivity of the graph. Graph topologies could thus be globally compared. In addition, the graph spectral domain could be used in order to define node modules in which edge-edge comparison can be performed. A combination of the resulting module-dependent scores could provide refined comparisons.

List of Figures

List of Tables

Bibliography

Abdulrehman, D., Monteiro, P. T., Teixeira, M. C., Mira, N. P., Lourenço, A. B., dos Santos, S. C., Cabrito, T. R., Francisco, A. P., Madeira, S. C., Aires, R. S., Oliveira, A. L., Sá-Correia, I., and Freitas, A. T. (2011). YEASTRACT: providing a programmatic access to curated transcriptional regulatory associations in *Saccharomyces cerevisiae* through a web services interface. *Nucleic Acids Res.*, **39**, D136–D140. Suppl. 1.

Aerts, S., Thijs, G., Coessens, B., Staes, M., Moreau, Y., and De Moor, B. (2003). Toucan: deciphering the cis-regulatory logic of coregulated genes. *Nucleic Acids Res.*, **31**(6), 1753–1764.

Aerts, S., Van Loo, P., Thijs, G., Mayer, H., de Martin, R., Moreau, Y., and De Moor, B. (2005). TOUCAN 2: the all-inclusive open source workbench for regulatory sequence analysis. *Nucleic Acids Res.*, **33**(Web Server), W393–W396.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Automat. Contr.*, **19**(6), 716–723.

Akutsu, T., Miyano, S., and Kuhara, S. (1999). Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In *Pac. Symp. Biocomput.*, volume 4, pages 17–28.

Altay, G. and Emmert-Streib, F. (2010). Inferring the conservative causal core of gene regulatory networks. *BMC Syst. Biol.*, **4**(1), 132.

Ambroise, C., Chiquet, J., and Matias, C. (2009). Inferring sparse Gaussian graphical models with latent structure. *Electron. J. Stat.*, **3**, 205–238.

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biol.*, **11**(10), R106.

Angulo, J. and Serra, J. (2003). Automatic analysis of DNA microarray images using mathematical morphology. *Bioinformatics*, **19**(5), 553–562.

Anscombe, F. J. (1948). The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, **35**(3/4), 246–254.

Arfin, S. M., Long, A. D., Ito, E. T., Tolleri, L., Riehle, M. M., Paegle, E. S., and Hatfield, G. W. (2000). Global gene expression profiling in *Escherichia coli* K12. The effects of integration host factor. *J. Biol. Chem.*, **275**(38), 29672–29684.

Aro, N., Saloheimo, A., Ilmen, M., and Penttila, M. (2001). ACEII, a novel transcriptional activator involved in regulation of cellulase and xylanase genes of *Trichoderma reesei*. *J. Biol. Chem.*, **276**(26), 24309–24314.

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. *Nat. Genet.*, **25**(1), 25–29.

Auer, P. L. and Doerge, R. W. (2011). A two-stage Poisson model for testing RNA-seq data. *Stat. Appl. Genet. Mol. Biol.*, **10**(1).

Ayasso, H. and Mohammad-Djafari, A. (2010). Joint NDT image restoration and segmentation using Gauss-Markov-Potts prior models and variational Bayesian computation. *IEEE Trans. Image Process.*, **19**(9), 2265–2277.

Azencott, C.-A., Grimm, D., Sugiyama, M., Kawahara, Y., and Borgwardt, K. M. (2013). Efficient network-guided multi-locus association mapping with graph cuts. *Bioinformatics*, **29**(13), i171–i179.

Babacan, S. D., Molina, R., and Katsaggelos, A. K. (2010). Sparse Bayesian image restoration. In *Proc. Int. Conf. Image Process.*, Hong Kong, China.

Babacan, S. D., Molina, R., and Katsaggelos, A. K. (2011). Variational Bayesian super resolution. *IEEE Trans. Image Process.*, **20**(4), 984–999.

Bailey, T. L., Williams, N., Misleh, C., and Li, W. W. (2006). MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res.*, **34**(Web Server), W369–W373.

Baird, D., Johnstone, P., and Wilson, T. (2004). Normalization of microarray data using a spatial mixed model analysis which includes splines. *Bioinformatics*, **20**(17), 3196–3205.

Balaji, S., Babu, M. M., Iyer, L. M., Luscombe, N. M., and Aravind, L. (2006). Comprehensive analysis of combinatorial regulation using the transcriptional regulatory network of yeast. *J. Mol. Biol.*, **360**(1), 213–227.

Baldi, P. and Long, A. D. (2001). A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics*, **17**(6), 509–519.

Bansal, M., Belcastro, V., Ambesi-Impiombato, A., and di Bernardo, D. (2007). How to infer gene networks from expression profiles. *Mol. Syst. Biol.*, **3**.

Bartlett, M. S. (1947). The use of transformations. *Biometrics*, **3**(1), 39–52.

Bauschke, H. H. and Combettes, P. L. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. CMS books in mathematics. Springer.

Beck, A. and Teboulle, M. (2009). Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Process.*, **18**(11), 2419–2434.

Beirlant, J., Dudewicz, E. J., Györfi, L., and van der Meulen, E. C. (1997). Nonparametric entropy estimation: An overview. *Int. J. Math. Stat. Sci.*, **6**, 17–39.

Bellot, P., Olsen, C., Salembier, P., Oliveras-Vergés, A., and Meyer, P. E. (2015). NetBenchmark: a bioconductor package for reproducible benchmarks of gene regulatory network inference. *BMC Bioinformatics*, **16**.

Ben-Dor, A., Shamir, R., and Yakhini, Z. (1999). Clustering gene expression patterns. *J. Comput. Biol.*, **6**(3-4), 281–297.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **57**(1), 289–300.

Berge, C. (1973). *Graphs and hypergraphs*. Elsevier.

Berger, J. A., Hautaniemi, S., Järvinen, A.-K., Edgren, H., Mitra, S. K., and Astola, J. (2004). Optimized LOWESS normalization parameter selection for DNA microarray data. *BMC Bioinformatics*.

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. John Wiley & Sons, Inc.

Besag, J. (1986). On the statistical analysis of dirty pictures. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **48**(3), 259–302.

Beucher, S. and Lantuéjoul, C. (1979). Use of watersheds in contour detection. In *Proc. Int. workshop image processing, real-time edge and motion detection/estimation*, Rennes, France.

Bioucas-Dias, J. (2006). Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors. *IEEE Trans. Image Process.*, **15**(4), 937–951.

Bioucas-Dias, J., Condessa, F., and Kovačević, J. (2014). Alternating direction optimization for image segmentation using hidden Markov measure field models. In K. O. Egiazarian, S. S. Agaian, and A. P. Gotchev, editors, *Proc. SPIE Image Process. Algorithms Syst.*, San Francisco, CA, USA. SPIE.

Blasi, M. F., Casorelli, I., Colosimo, A., Blasi, F. S., Bignami, M., and Giuliani, A. (2005). A recursive network approach can identify constitutive regulatory circuits in gene expression data. *Phys. Stat. Mech. Appl.*, **348**, 349–370.

Bolstad, B. M., Irizarry, R. A., Åstrand, M., and Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, **19**(2), 185–193.

Bondy, A. and Murty, U. S. R. (2007). *Graph Theory*. Springer.

Bonneau, R., Reiss, D., Shannon, P., Facciotti, M., Hood, L., Baliga, N., and Thorsson, Y. (2006). The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets *de novo*. *Genome Biol.*, **7**(5), R36+.

Boykov, Y. and Jolly, M.-P. (2000). Interactive organ segmentation using graph cuts. In *Proc. Medical Image Computing Computer-Assisted Intervention Conf.*, pages 276–286. Springer.

Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(9), 1124–1137.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, **23**(11), 1222–1239.

Breiman, L. (2001). Random forests. *Mach. Learn.*, **45**(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, CA, USA.

Briceño-Arias, L. M., Combettes, P. L., Pesquet, J.-C., and Pustelnik, N. (2011). Proximal algorithms for multicomponent image processing. *J. Math. Imaging Vision*, **41**(1), 3–22.

Bullard, J. H., Purdom, E., Hansen, K. D., and Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, **11**(1), 94.

Butte, A. J. and Kohane, I. S. (2000). Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. In R. B. Altman, A. K. Dunker, L. Hunter, K. Lauderdale, and T. E. Klein, editors, *Pac. Symp. Biocomput.*, volume 5, pages 415–429, Hawaii, HI, USA.

Cai, X. (2015). Variational image segmentation model coupled with image restoration achievements. *Pattern Recogn.*, **48**(6), 2029–2042.

Cai, X., Chan, R., and Zeng, T. (2013). A two-stage image segmentation method using a convex variant of the Mumford-Shah model and thresholding. *SIAM J. Imaging Sci.*, **6**(1), 368–390.

Callow, M. J., Dudoit, S., Gong, E. L., Speed, T. P., and Rubin, E. M. (2000). Microarray expression profiling identifies genes with altered expression in HDL-deficient mice. *Genome Res.*, **10**(12), 2022–2029.

Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**(6), 679–698.

Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Duxbury, 2nd edition.

Chaâri, L., Pustelnik, N., Chaux, C., and Pesquet, J.-C. (2009). Solving inverse problems with overcomplete transforms and convex optimization techniques. In V. K. Goyal, M. Papadakis, and D. Van De Ville, editors, *Proc. SPIE, Wavelets*, volume 7446, San Diego, CA, USA.

Chaari, L., Forbes, F., Vincent, T., Dojat, M., and Ciuciu, P. (2011). Variational solution to the joint detection estimation of brain activity in fMRI. In G. Fichtinger, A. Martel, and T. Peters, editors, *Proc. Medical Image Computing Computer-Assisted Intervention Conf.*, volume 6892 of *Lect. Notes Comput. Sci.*, pages 260–268.

Chai, L. E., Loh, S. K., Low, S. T., Mohamad, M. S., Deris, S., and Zakaria, Z. (2014). A review on the computational approaches for gene regulatory network construction. *Comput. Biol. Med.*, **48**, 55–65.

Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision*, **40**(1), 120–145.

Chantas, G., Galatsanos, N., Likas, A., and Saunders, M. (2008). Variational Bayesian image restoration based on a product of t-distributions image prior. *IEEE Trans. Image Process.*, **17**(10), 1795–1805.

Charbonnier, C., Chiquet, J., and Ambroise, C. (2010). Weighted-Lasso for structured network inference from time course data. *Stat. Appl. Genet. Mol. Biol.*, **9**(1).

Chaux, C., Combettes, P. L., Pesquet, J.-C., and Wajs, V. R. (2007). A variational formulation for frame based inverse problems. *Inverse Problems*, **23**(4), 1495–1518.

Chaux, C., Duval, L., Benazza-Benyahia, A., and Pesquet, J.-C. (2008). A nonlinear Stein based estimator for multichannel image denoising. *IEEE Trans. Signal Process.*, **56**(8), 3855–3870.

Chaux, C., Benazza-Benyahia, A., Pesquet, J.-C., and Duval, L. (2009). Wavelet transform for the denoising of multivariate images. In J. Chanussot, C. Collet, and K. Chehdi, editors, *Multivariate Image Processing*, pages 203–238. ISTE Ltd and John Wiley & Sons Inc.

Chen, T., He, H. L., and Church, G. M. (1999). Modeling gene expression with differential equations. In *Pac. Symp. Biocomput.*

Chen, Y., Dougherty, E. R., and Bittner, M. L. (1997). Ratio-based decisions and the quantitative analysis of cDNA microarray images. *J. Biomed. Opt.*, **2**(4), 364.

Chen, Y., Li, Y., Narayan, R., Subramanian, A., and Xie, X. (2016). Gene expression inference with deep learning. *Bioinformatics*, **32**(12), 1832–1839.

Chen, Z., Babacan, S. D., Molina, R., and Katsaggelos, A. K. (2014). Variational Bayesian methods for multimedia problems. *IEEE Trans. Multimedia*, **16**(4), 1000–1017.

Cheng, H. D., Jiang, X. H., Sun, Y., and Wang, J. (2001). Color image segmentation: advances and prospects. *Pattern Recogn.*, **34**(12), 2259–2281.

Cherry, J. M., Hong, E. L., Amundsen, C., Balakrishnan, R., Binkley, G., Chan, E. T., Christie, K. R., Costanzo, M. C., Dwight, S. S., Engel, S. R., Fisk, D. G., Hirschman, J. E., Hitz, B. C., Karra, K., Krieger, C. J., Miyasato, S. R., Nash, R. S., Park, J., Skrzypek, M. S., Simison, M., Weng, S., and Wong, E. D. (2012). Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic Acids Res.*, **40**(Database-Issue), D700–D705.

Chierchia, G., Pustelnik, N., Pesquet-Popescu, B., and Pesquet, J.-C. (2014). A nonlocal structure tensor-based approach for multicomponent image recovery problems. *IEEE Trans. Image Process.*, **23**(12), 5531–5544.

Chiquet, J. (2015). *Contributions to Sparse Methods for Complex Data Analysis*. Habilitation à diriger des recherches (HDR), Université d'Évry-Val-d'Essonne.

Chiquet, J., Smith, A., Grasseau, G., Matias, C., and Ambroise, C. (2009). SIMoNe: Statistical Inference for MOdular NEtworks. *Bioinformatics*, **25**(3), 417–418.

Chiquet, J., Grandvalet, Y., and Charbonnier, C. (2012). Sparsity in sign-coherent groups of variables via the cooperative-lasso. *Ann. Appl. Stat.*, **6**(2), 795–830.

Choudrey, R. A. (2002). *Variational Methods for Bayesian Independent Component Analysis*. Ph.D. thesis, University of Oxford.

Chouzenoux, E., Idier, J., and Moussaoui, S. (2011). A majorize-minimize strategy for subspace optimization applied to image restoration. *IEEE Trans. Image Process.*, **20**(6), 1517–1528.

Chouzenoux, E., Jezierska, A., Pesquet, J.-C., and Talbot, H. (2013). A majorize-minimize subspace approach for ℓ_2-ℓ_0 image regularization. *SIAM J. Imaging Sci.*, **6**(1), 563–591.

Chouzenoux, E., Pesquet, J.-C., and Repetti, A. (2014). Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *J. Optim. Theory Appl.*, **162**(1), 107–132.

Chouzenoux, E., Pesquet, J.-C., and Repetti, A. (2016). A block coordinate variable metric forward-backward algorithm. *J. Global Optim.*, **66**(3), 457–485.

Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Rev.*, **51**(4), 661–703.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.*, **74**(368), 829–836.

Combettes, P. L. and Pesquet, J.-C. (2011). Proximal splitting methods in signal processing. In H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer Verlag.

Combettes, P. L. and Wajs, V. R. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, **4**(4), 1168–1200.

Coradetti, S. T., Craig, J. P., Xiong, Y., Shock, T., Tian, C., and Glass, N. L. (2012). Conserved and essential transcription factors for cellulase gene expression in ascomycete fungi. *Proc. Nat. Acad. Sci. U.S.A.*, **109**(19), 7397–7402.

Couprie, C., Grady, L., Najman, L., and Talbot, H. (2011). Power watershed: A unifying graph-based optimization framework. *IEEE Trans. Pattern Anal. Mach. Intell.*, **33**(7), 1384–1399.

Cowley, M. J., Pinese, M., Kassahn, K. S., Waddell, N., Pearson, J. V., Grimmond, S. M., Biankin, A. V., Hautaniemi, S., and Wu, J. (2011). PINA v2.0: mining interactome modules. *Nucleic Acids Res.*, **40**(D1), D862–D865.

Crick, F. (1970). Central dogma of molecular biology. *Nature*, **227**(5258), 561–563.

Cui, X. and Churchill, G. A. (2003). Statistical tests for differential expression in cDNA microarray experiments. *Genome Biol.*, **4**(4), 210.

Darbon, J. (2009). Global optimization for first order Markov Random Fields with submodular priors. *Discrete Appl. Math.*, **157**(16), 3412–3423.

Daub, C. O., Steuer, R., Selbig, J., and Kloska, S. (2004). Estimating mutual information using B-spline functions—an improved similarity measure for analysing gene expression data. *BMC Bioinformatics*, **5**(1), 118.

Davis, J. and Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In *Proc. Int. Conf. Mach. Learn.* Association for Computing Machinery (ACM).

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *J. Comput. Biol.*, **9**(1), 67–103.

De Smet, R. and Marchal, K. (2010). Advantages and limitations of current network inference methods. *Nat. Rev. Microbiol.*, **8**(10), 717–729.

de Souto, M. C. P., Costa, I. G., de Araujo, D. S. A., Ludermir, T. B., and Schliep, A. (2008). Clustering cancer gene expression data: a comparative study. *BMC Bioinformatics*, **9**(1), 497.

Dempster, A. P. (1972). Covariance selection. *Biometrics*, **28**(1), 157–175.

Denton, J. A. and Kelly, J. M. (2011). Disruption of *Trichoderma reesei* cre2, encoding an ubiquitin c-terminal hydrolase, results in increased cellulase activity. *BMC Biotechnol.*, **11**(1), 103.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numer. Math.*, **1**, 269–271.

Dillies, M.-A., Rau, A., Aubert, J., Hennequet-Antier, C., Jeanmougin, M., Servant, N., Keime, C., Marot, G., 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. *Brief. Bioinform.*, **14**(6), 671–683.

Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *J. Bioinformatics Comput. Biol.*, **3**(2), 185–205.

Dojer, N., Gambin, A., and Tiuryn, J. (2006). Applying dynamic Bayesian networks to perturbed gene expression data. *BMC Bioinformatics*, **7**, 249.

Dong, X., Frossard, P., Vandergheynst, P., and Nefedov, N. (2012). Clustering with multi-layer graphs: A spectral perspective. *IEEE Trans. Signal Process.*, **60**(11), 5820–5831.

Dong, X., Thanou, D., Frossard, P., and Vandergheynst, P. (2016). Learning Laplacian matrix in smooth graph signal representations. *IEEE Trans. Signal Process.*, **64**(23), 6160–6173.

Donoho, D. L., Elad, M., and Temlyakov, V. N. (2006). Stable recovery of sparse over-complete representations in the presence of noise. *IEEE Trans. Inform. Theory*, **52**(1), 6–18.

Dougherty, E. R., Shmulevich, I., Chen, J., and Wang, Z. J., editors (2005). *Genomic Signal Processing and Statistics*. Hindawi Publishing Corporation.

Dudoit, S., Yang, Y. H., Callow, M. J., and Speed, T. P. (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statist. Sinica*, **12**, 111–139.

Dunn, O. J. (1959). Estimation of the medians for dependent variables. *Ann. Math. Statist.*, **30**(1), 192–197.

Dunn, O. J. (1961). Multiple comparisons among means. *J. Am. Stat. Assoc.*, **56**(293), 52–64.

Dupé, F.-X., Fadili, J. M., and Starck, J.-L. (2009). A proximal iteration for deconvolving poisson noisy images using sparse representations. *IEEE Trans. Image Process.*, **18**(2), 310–321.

Durbin, B. P., Hardin, J. S., Hawkins, D. M., and Rocke, D. M. (2002). A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics*, **18**(Suppl. 1), S105–S110.

Edmonds, J. and Karp, R. M. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, **19**(2), 248–264.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, **7**(1), 1–26.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *J. Am. Stat. Assoc.*, **78**(382), 316–331.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Ann. Statist.*, **32**(2), 407–499.

Eisenberg, E. and Levanon, E. Y. (2013). Human housekeeping genes, revisited. *Trends Genet.*, **29**(10), 569–574.

Elbakri, I. A. and Fessler, J. A. (2003). Segmentation-free statistical image reconstruction for polyenergetic x-ray computed tomography with experimental validation. *Phys. Med. Biol.*, **48**(15), 2453–2477.

Emmert-Streib, F., Glazko, G. V., Altay, G., and de Matos Simoes, R. (2012). Statistical inference and reverse engineering of gene regulatory networks from observational expression data. *Front. Genet.*, **3**.

Espinosa-Soto, C. and Wagner, A. (2010). Specialization can drive the evolution of modularity. *PLoS Comput. Biol.*, **6**(3), e1000719.

Evans, C., Hardin, J., and Stoebel, D. M. (2017). Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions. *Brief. Bioinform.*, pages 1–17. n.

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J. J., and Gardner, T. S. (2007). Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS Biol.*, **5**(1), 54–66.

Fan, J., Liao, Y., and Liu, H. (2016). An overview of the estimation of large covariance and precision matrices. *Econom. J.*, **19**(1), C1–C32.

Feizi, S., Marbach, D., Médard, M., and Kellis, M. (2013). Network deconvolution as a general method to distinguish direct dependencies in networks. *Nat. Biotechnol.*, **31**(8), 726–733.

Figueiredo, M. A. T. (2003). An EM algorithm for wavelet-based image restoration. *IEEE Trans. Image Process.*, **12**(8), 906–916.

Filkov, V. (2005). Identifying gene regulatory networks from gene expression data. In S. Aluru, editor, *Handbook of Computational Molecular Biology*, Computer & Information Science Series, chapter 27. Chapman & Hall/CRC.

Ford, Jr., L. R. and Fulkerson, D. R. (1956). Maximal flow through a network. *Canad. J. Math.*, **8**, 399–404.

Foreman, P. K., Brown, D., Dankmeyer, L., Dean, R., Diener, S., Dunn-Coleman, N. S., Goedegebuur, F., Houfek, T. D., England, G. J., Kelley, A. S., Meerman, H. J., Mitchell, T., Mitchinson, C., Olivares, H. A., Teunissen, P. J., Yao, J., and Ward, M. (2003). Transcriptional regulation of biomass-degrading enzymes in the filamentous fungus *Trichoderma reesei*. *J. Biol. Chem.*, **278**.

Fortunato, S. (2010). Community detection in graphs. *Phys. Rep.*, **486**, 75–174.

Franceschini, A., Szklarczyk, D., Frankild, S., Kuhn, M., Simonovic, M., Roth, A., Lin, J., Minguez, P., Bork, P., von Mering, C., and Jensen, L. J. (2013). STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res.*, **41**, D808–D815.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, **7**(3-4), 601–620.

Fu, W. J. (1998). Penalized regressions: The bridge versus the lasso. *J. Comput. Graph. Stat.*, **7**(3), 397–416.

Fujita, A., Sato, J. R., de Oliveira Rodrigues, L., Ferreira, C. E., and Sogayar, M. C. (2006). Evaluating different methods of microarray data normalization. *BMC Bioinformatics*, **7**(1), 469.

Gadaleta, F. (2015). Are we far from correctly inferring gene interaction networks with Lasso? *PREPRINT*.

Gama-Castro, S., Jiménez-Jacinto, V., Peralta-Gil, M., Santos-Zavaleta, A., Peñaloza Spinola, M. I., Contreras-Moreira, B., Segura-Salazar, J., Muñiz Rascado, L., Martínez-Flores, I., Salgado, H., Bonavides-Martínez, C., Abreu-Goodger, C., Rodríguez-Penagos, C., Miranda-Ríos, J., Morett, E., Merino, E., Huerta, A. M., Treviño-Quintanilla, L., and Collado-Vides, J. (2008). RegulonDB (version 6.0): gene regulation model of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic Acids Res.*, **36**(Database), D120–D124.

Gama-Castro, S., Salgado, H., Peralta-Gil, M., Santos-Zavaleta, A., Muñiz Rascado, L., Solano-Lira, H., Jimenez-Jacinto, V., Weiss, V., García-Sotelo, J. S., López-Fuentes, A., Porrón-Sotelo, L., Alquicira-Hernández, S., Medina-Rivera, A., Martínez-Flores, I., K., A.-H., Martínez-Adame, R., Bonavides-Martínez, C., Miranda-Ríos, J., Huerta, A. M., Mendoza-Vargas, A., Collado-Torres, L., Taboada, B., Vega-Alvarado, L., Olvera, M.,

Olvera, L., Grande, R., Morett, E., and Collado-Vides, J. (2011). RegulonDB version 7.0: transcriptional regulation of *Escherichia coli* K-12 integrated within genetic sensory response units (gensor units). *Nucleic Acids Res.*, **39**(Database), D98–D105.

Gama-Castro, S., Salgado, H., Santos-Zavaleta, A., Ledezma-Tejeida, D., Muñiz-Rascado, L., García-Sotelo, J. S., Alquicira-Hernández, K., Martínez-Flores, I., Pannier, L., Castro-Mondragón, J. A., Medina-Rivera, A., Solano-Lira, H., Bonavides-Martínez, C., Pérez-Rueda, E., Alquicira-Hernández, S., Porrón-Sotelo, L., López-Fuentes, A., Hernández-Koutoucheva, A., Del Moral-Chávez, V., Rinaldi, F., and Collado-Vides, J. (2016). RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Res.*, **44**(D1), D133–D143.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Mach. Learn.*, **63**(1), 3–42.

Gierliński, M., Cole, C., Schofield, P., Schurch, N. J., Sherstnev, A., Singh, V., Wrobel, N., Gharbi, K., Simpson, G., Owen-Hughes, T., Blaxter, M., and Barton, G. J. (2015). Statistical models for RNA-seq data derived from a two-condition 48-replicate experiment. *Bioinformatics*, **31**(22), 3625–3630.

Gilboa, G. and Osher, S. (2009). Nonlocal operators with applications to image processing. *Multiscale Model. Simul.*, **7**(3), 1005–1028.

Goldberg, A. V. and Tarjan, R. E. (1986). A new approach to the maximum flow problem. In *Proc. ACM Symp. Theor. Comput.*, pages 136–146, Berkeley, CA, USA.

Gómez, E., Gómez-Villegas, M. A., and Marín, J. M. (1998). A multivariate generalization of the power exponential family of distributions. *Commun. Stat. Theory Methods*, **27**(3), 589–600.

Gómez-Sánchez-Manzano, E., Gómez-Villegas, M. A., and Marín, J. M. (2008). Multivariate exponential power distributions as mixtures of normal distributions with Bayesian applications. *Commun. Stat. Theory Methods*, **37**(6), 972–985.

Grady, L. (2006). Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**(11), 1768–1783.

Grady, L. J. and Polimeni, J. R. (2010). *Discrete calculus: Applied analysis on graphs for computational science*. Springer.

Grossetête, S., Labedan, B., and Lespinet, O. (2010). FUNGIpath: a tool to assess fungal metabolic pathways predicted by orthology. *BMC Genom.*, **11**(1), 81.

Günther, F., Wawro, N., and Bammann, K. (2009). Neural networks for modeling gene-gene interactions in association studies. *BMC Genet.*, **10**(1), 87.

Ha, M. J., Baladandayuthapani, V., and Do, K.-A. (2015). DINGO: differential network analysis in genomics. *Bioinformatics*, **31**(21), 3413–3420.

Hadamard, J. (1902). Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, **13**, 49–52.

Häkkinen, M., Valkonen, M. J., Westerholm-Parvinen, A., Aro, N., Arvas, M., Vitikainen, M., Penttilä, M., Saloheimo, M., and Pakula, T. M. (2014). Screening of candidate regulators for cellulase and hemicellulase production in *Trichoderma reesei* and identification of a factor essential for cellulase production. *Biotechnol. Biofuels*, **7**(1), 14.

Halleran, A., Clamons, S., and Saha, M. (2015). Transcriptomic characterization of an infection of *Mycobacterium smegmatis* by the cluster A4 mycobacteriophage Kampy. *PLoS One*, **10**(10), e0141100.

Hardcastle, T. J. and Kelly, K. A. (2010). baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics*, **11**(1), 1–14.

Hastie, T., Tibshirani, R., and Friedman, J. (2013). *The elements of statistical learning*. Springer.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press.

Haury, A.-C., Mordelet, F., Vera-Licona, P., and Vert, J.-P. (2012). TIGRESS: Trustful Inference of Gene REgulation using Stability Selection. *BMC Syst. Biol.*, **6**(1), 145.

He, Y., Hussaini, M. Y., Ma, J., Shafei, B., and Steidl, G. (2012). A new fuzzy c-means method with total variation regularization for segmentation of images with noisy and incomplete data. *Pattern Recogn.*, **45**(9), 3463–3471.

Hecker, M., Lambeck, S., Toepfer, S., van Someren, E., and Guthke, R. (2009). Gene regulatory network inference: Data integration in dynamic models—a review. *BioSystems*, **96**(1), 86–103.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.*, **20**(3), 197–243.

Hess, K. R., Anderson, K., Symmans, W. F., Valero, V., Ibrahim, N., Mejia, J. A., Booser, D., Theriault, R. L., Buzdar, A. U., Dempsey, P. J., Rouzier, R., Sneige, N., Ross, J. S., Vidaurre, T., Gómez, H. L., Hortobagyi, G. N., and Pusztai, L. (2006). Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *J. Clin. Oncol.*, **24**(26), 4236–4244.

Hicks, S. C. and Irizarry, R. A. (2015). *quantro*: a data-driven approach to guide the choice of an appropriate normalization method. *Genome Biol.*, **16**(1), 117.

Horimoto, K. and Toh, H. (2001). Statistical estimation of cluster boundaries in gene expression profile data. *Bioinformatics*, **17**(12), 1143–1151.

Howe, E., Holton, K., Nair, S., Schlauch, D., Sinha, R., and Quackenbush, J. (2010). MeV: MultiExperiment Viewer. In F. M. Ochs, T. J. Casagrande, and V. R. Davuluri, editors, *Biomedical Informatics for Cancer Research*, pages 267–277. Springer, Boston, MA.

Hu, Z., Killion, P. J., and Iyer, V. R. (2007). Genetic reconstruction of a functional transcriptional regulatory network. *Nat. Genet.*, **39**(5), 683–687.

Huber, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Statist.*, **35**(1), 73–101.

Huber, P. J. and Ronchetti, E. M. (2009). *Robust statistics*. Wiley, 2nd edition.

Huber, W., von Heydebreck, A., Sultmann, H., Poustka, A., and Vingron, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, **18**(Suppl. 1), 96–104.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *J. Classif.*, **2**(1), 193–218.

Hunter, D. R. and Lange, K. (2004). A tutorial on MM algorithms. *Am. Stat.*, **58**(1), 30–37.

Huynh-Thu, V. A. and Sanguinetti, G. (2015). Combining tree-based and dynamical systems for the inference of gene regulatory networks. *Bioinformatics*, **31**(10), 1614–1622.

Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., and Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. *PLoS One*, **5**(9), e12776.

Ideker, T., Thorsson, V., and Karp, R. (2000). Discovery of regulatory interactions through perturbation: inference and experimental design. In *Pac. Symp. Biocomput.*, volume 5, pages 302–313.

Izenman, A. J. (2008). *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice Hall.

Jezierska, A., Chouzenoux, E., Pesquet, J.-C., and Talbot, H. (2012). A primal-dual proximal splitting approach for restoring data corrupted with Poisson-Gaussian noise. In *Proc. Int. Conf. Acoust. Speech Signal Process.*, Kyoto, Japan.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Mach. Learn.*, **37**(2), 183–233.

Joshi, A., De Smet, R., Marchal, K., Van de Peer, Y., and Michoel, T. (2009). Module networks revisited: computational assessment and prioritization of model predictions. *Bioinformatics*, **25**(4), 490–496.

Jourdier, E., Cohen, C., Poughon, L., Larroche, C., Monot, F., and Chaabane, F. (2013). Cellulase activity mapping of *Trichoderma reesei* cultivated in sugar mixtures under fed-batch conditions. *Biotechnol. Biofuels*, **6**(1), 79.

Kaderali, L. and Radde, N. (2008). Inferring gene regulatory networks from expression data. In A. Kelemen, A. Abraham, and Y. Chen, editors, *Computational Intelligence in Bioinformatics*, pages 33–74. Springer, Berlin, Heidelberg.

Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, **28**(1), 27–30.

Kauffman, S. (1969). Homeostasis and differentiation in random genetic control networks. *Nature*, **224**(5215), 177–178.

Kaufman, L. and Rousseeuw, P. J. (2005). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience.

Kerr, M. K., Martin, M., and Churchill, G. A. (2000). Analysis of variance for gene expression microarray data. *J. Comput. Biol.*, **7**(6), 819–837.

Keseler, I. M., Mackie, A., Peralta-Gil, M., Santos-Zavaleta, A., Gama-Castro, S., Bonavides-Martínez, C., Fulcher, C., Huerta, A. M., Kothari, A., Krummenacker, M., Latendresse, M., Muñiz Rascado, L., Ong, Q., Paley, S., Schröder, I., Shearer, A. G., Subhraveti, P., Travers, M., Weerasinghe, D., Weiss, V., Collado-Vides, J., Gunsalus, R. P., Paulsen, I., and Karp, P. D. (2013). EcoCyc: fusing model organism databases with systems biology. *Nucleic Acids Res.*, **41**(D1), D605–D612.

Kharchenko, P. V., Silberstein, L., and Scadden, D. T. (2014). Bayesian approach to single-cell differential expression analysis. *Nat. Meth.*, **11**(7), 740–742.

Kim, Y., Choi, H., and Oh, H.-S. (2008). Smoothly clipped absolute deviation on high dimensions. *J. Am. Stat. Assoc.*, **103**(484), 1665–1673.

Klamt, S., Haus, U.-U., and Theis, F. (2009). Hypergraphs and cellular networks. *PLoS Comput. Biol.*, **5**(5), e1000385.

Kohane, I. S., Kho, A. T., and Butte, A. J. (2003). *Microarrays for an Integrative Genomics*. The MIT Press.

Kohli, P., Ladický, L., and Torr, P. H. S. (2009). Robust higher order potentials for enforcing label consistency. *Int. J. Comput. Vis.*, **82**(3), 302–324.

Kohonen, T. (2000). *Self-Organizing Maps*. Springer, 3rd edition.

Kolmogorov, V. and Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(2), 147–159.

Komodakis, N. and Pesquet, J.-C. (2015). Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Signal Process. Mag.*, **32**(6), 31–54.

Komodakis, N., Paragios, N., and Tziritas, G. (2011). MRF energy minimization and beyond via dual decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, **33**(3), 531–552.

Küffner, R., Petri, T., Tavakkolkhah, P., Windhager, L., and Zimmer, R. (2012). Inferring gene regulatory networks by ANOVA. *Bioinformatics*, **28**(10), 1376–1382.

Kurt, Z., Aydin, N., and Altay, G. (2014). A comprehensive comparison of association estimators for gene network inference algorithms. *Bioinformatics*, **30**(15), 2142–2149.

Langfelder, P. and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, **9**(1), 559.

Lantéri, H. and Theys, C. (2005). Restoration of astrophysical images—the case of Poisson data with additive Gaussian noise. *EURASIP J. Adv. Signal Process.*, **2005**(15), 2500–2513.

Lauritzen, S. L. (1996). *Graphical Models*. Oxford Statistical Science Series. Oxford University Press.

Lawson, J. D. and Lim, Y. (2001). The geometric mean, matrices, metrics, and more. *Amer. Math. Monthly*, **108**(9), 797–812.

Lee, W.-P. and Yang, K.-C. (2008). A clustering-based approach for inferring recurrent neural networks as gene regulatory networks. *Neurocomputing*, **71**(4-6), 600–610.

Leng, N., Dawson, J. A., Thomson, J. A., Ruotti, V., Rissman, A. I., Smits, B. M. G., Haag, J. D., Gould, M. N., Stewart, R. M., and Kendziorski, C. (2013). EBSeq: an empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics*, **29**(8), 1035–1043.

Lévy-Leduc, C., Delattre, M., Mary-Huard, T., and Robin, S. (2014). Two-dimensional segmentation for analyzing HiC data. *Bioinformatics*.

Li, H. and Gui, J. (2005). Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, **7**(2), 302–317.

Liang, K. and Keleş, S. (2012). Normalization of ChIP-seq data with control. *BMC Bioinformatics*, **13**(1), 199.

Liang, K.-C. and Wang, X. (2008). Gene regulatory network reconstruction using conditional mutual information. *EURASIP J. Bioinformatics Syst. Biol.*, **2008**, 1–14.

Liang, S., Fuhrman, S., and Somogyi, R. (1998). REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Pac. Symp. Biocomput.*, volume 3, pages 18–29.

Likas, C. L. and Galatsanos, N. P. (2004). A variational approach for Bayesian blind image deconvolution. *IEEE Trans. Signal Process.*, **52**(8), 2222–2233.

Lim, W. K., Wang, K., Lefebvre, C., and Califano, A. (2007). Comparative analysis of microarray normalization procedures: effects on reverse engineering gene networks. *Bioinformatics*, **23**(13), i282–i288.

Lin, Y., Golovnina, K., Chen, Z.-X., Lee, H. N., Negron, Y. L. S., Sultana, H., Oliver, B., and Harbison, S. T. (2016). Comparison of normalization and differential expression analyses using RNA-seq data from 726 individual *Drosophila melanogaster*. *BMC Genom.*, **17**.

Lindöf, A. and Olsson, B. (2003). Genetic network inference: the effects of preprocessing. *BioSystems*, **72**(3), 229–239.

Liu, L.-Z., Wu, F.-X., and Zhang, W.-J. (2014). A group LASSO-based method for robustly inferring gene regulatory networks from multiple time-course datasets. *BMC Syst. Biol.*, **8**(Suppl. 3), S1.

Liu, Z.-P. (2015). Reverse engineering of genome-wide gene regulatory networks from gene expression data. *Curr. Genom.*, **16**, 3–22.

Lönnstedt, I. and Speed, T. (2002). Replicated microarray data. *Statist. Sinica*, **12**(1), 31–46.

Lu, T., Liang, H., Li, H., and Wu, H. (2011). High-dimensional ODEs coupled with mixed-effects modeling techniques for dynamic gene regulatory network identification. *J. Am. Stat. Assoc.*, **106**(496), 1242–1258.

Luo, W., Hankenson, K. D., and Woolf, P. J. (2008). Learning transcriptional regulatory networks from high throughput gene expression data using continuous three-way mutual information. *BMC Bioinformatics*, **9**(1), 467.

Luo, Z. Q. and Tseng, P. (1992). On the convergence of the coordinate descent method for convex differentiable minimization. *J. Optim. Theory Appl.*, **72**(1), 7–35.

Macaulay, F. R. (1931). The Whittaker-Henderson method of graduation. In *The Smoothing of Time Series*, pages 89–99. National Bureau of Economic Research.

Mach-Aigner, A. R., Pucher, M. E., Steiger, M. G., Bauer, G. E., Preis, S. J., and Mach, R. L. (2008). Transcriptional regulation of xyr1, encoding the main regulator of the xylanolytic and cellulolytic enzyme system in *Hypocrea jecorina*. *Appl. Environ. Microbiol.*, **74**(21), 6554–6562.

MacIsaac, K. D., Wang, T., Gordon, D. B., Gifford, D. K., Stormo, G. D., and Fraenkel, E. (2006). An improved map of conserved regulatory sites for *Saccharomyces cerevisiae*. *BMC Bioinformatics*, **7**(1), 113.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. Math. Statist. Prob.*, volume 1: Statistics, pages 281–297.

Man, B. D., Nuyts, J., Dupont, P., Marchal, G., and Suetens, P. (2001). An iterative maximum-likelihood polychromatic algorithm for CT. *IEEE Trans. Med. Imag.*, **20**(10), 999–1008.

Marbach, D., Schaffter, T., Mattiussi, C., and Floreano, D. (2009). Generating realistic *in silico* gene networks for performance assessment of reverse engineering methods. *J. Comput. Biol.*, **16**(2), 229–239.

Marbach, D., Prill, R. J., Schaffter, T., Mattiussi, C., Floreano, D., and Stolovitzky, G. (2010). Revealing strengths and weaknesses of methods for gene network inference. *Proc. Nat. Acad. Sci. U.S.A.*, **107**(14), 6286–6291.

Marbach, D., Costello, J. C., Küffner, R., Vega, N. M., Prill, R. J., Camacho, D. M., Allison, K. R., The DREAM5 Consortium, Kellis, M., Collins, J. J., and Stolovitzky, G. (2012). Wisdom of crowds for robust gene network inference. *Nat. Meth.*, **9**(8), 796–804.

Margolin, A. 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)**(5), S7.

Marinari, E. and Marra, R. (1990). Cluster algorithms for the generalized $3d$, $3q$ Potts model. *Nucl. Phys. B*, **342**(3), 737–752.

Marnissi, Y., Chouzenoux, E., Pesquet, J.-C., and Benazza-Benyahia, A. (2016). An auxiliary variable method for Langevin based MCMC algorithms. In *Proc. IEEE Workshop Stat. Signal Process.*, Palma de Majorca, Spain.

Mathelier, A., Zhao, X., Zhang, A. W., Parcy, F., Worsley-Hunt, R., Arenillas, D. J., Buchman, S., Chen, C.-Y., Chou, A., Ienasescu, H., Lim, J., Shyr, C., Tan, G., Zhou, M., Lenhard, B., and Sandelin, A. (2013). JASPAR 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. *Nucleic Acids Res.*, **42**(D1), D142–D147.

McGrory, C. A., Titterington, D. M., Reeves, R., and Pettitt, A. N. (2009). Variational Bayes for estimating the parameters of a hidden Potts model. *Stat. Comput.*, **19**(3), 329–340.

McLachlan, G. J. and Krishnan, T. (2008). *The EM Algorithm and Extensions*. Wiley, 2nd edition.

Meilă, M. (2007). Comparing clusterings—an information based distance. *J. Multivariate Anal.*, **98**(5), 873–895.

Meilă, M. (2003). Comparing clusterings by the variation of information. In B. Schölkopf and M. K. Warmuth, editors, *Learning Theory and Kernel Machines*, volume 2777 of *Lect. Notes Comput. Sci.*, pages 173–187. Springer.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *Ann. Statist.*, **34**(3), 1436–1462.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **72**(4), 417–473.

Merris, R. (2000). *Graph Theory*. Series in Discrete Mathematics and Optimization. Wiley.

Meyer, F. (1994). Minimum spanning forests for morphological segmentation. In J. Serra and P. Soille, editors, *Mathematical Morphology and Its Applications to Image Processing*, pages 77–84. Springer, Dordrecht.

Meyer, P. E., Kontos, K., Lafitte, F., and Bontempi, G. (2007). Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J. Bioinformatics Syst. Biol.*, **2007**, 1–9.

Meyer, P. E., Lafitte, F., and Bontempi, G. (2008). minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information. *Bioinformatics*, **9**, 461+.

Min, S., Lee, B., and Yoon, S. (2016). Deep learning in bioinformatics. *Brief. Bioinform.*, page bbw068.

Molina, R. (1994). On the hierarchical Bayesian approach to image restoration: applications to astronomical images. *IEEE Trans. Pattern Anal. Mach. Intell.*, **16**(11), 1122–1128.

Molina, R., Katsaggelos, A. K., and Mateos, J. (1999). Bayesian and regularization methods for hyperparameter estimation in image restoration. *IEEE Trans. Image Process.*, **8**(2), 231–246.

Montenecourt, B. S. and Eveleigh, D. E. (1977). Preparation of mutants of *Trichoderma reesei* with enhanced cellulase production. *Appl. Environ. Microbiol.*, **34**(6), 777—782.

Mordelet, F. and Vert, J.-P. (2008). SIRENE: supervised inference of regulatory networks. *Bioinformatics*, **24**(16), i76–i82.

Moreau, J. J. (1965). Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, **93**, 273–299.

Mortazavi, A., Williams, B., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Meth.*, **5**(7), 621–628.

Nakari-Setälä, T., Paloheimo, M., Kallio, J., Vehmaanperä, J., Penttilä, M., and Saloheimo, M. (2009). Genetic modification of carbon catabolite repression in *Trichoderma reesei* for improved protein production. *Appl. Environ. Microbiol.*, **75**(14), 4853–4860.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Comput. J.*, **7**(4), 308–313.

Newman, M. E. J. (2012). Communities, modules and large-scale structure in networks. *Nat. Phys.*, **8**(1), 25–31.

Newton, M. A., Kendziorski, C. M., Richmond, C. S., Blattner, F. R., and Tsui, K. W. (2001). On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data. *J. Comput. Biol.*, **8**(1), 37–52.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, pages 849–856. MIT Press.

Ning, X., Selesnick, I. W., and Duval, L. (2014). Chromatogram baseline estimation and denoising using sparsity (BEADS). *Chemometr. Intell. Lab. Syst.*, **139**, 156–167.

Obozinski, G., Jacob, L., and Vert, J.-P. (2011). Group lasso with overlaps: the latent group lasso approach. Technical report.

O'Connor, D. and Vandenberghe, L. (2017). Total variation image deblurring with space-varying kernel via Douglas-Rachford splitting. *Comput. Optim. Appl.*

Okawa, S., Angarica, V. E., Lemischka, I., Moore, K., and del Sol, A. (2015). A differential network analysis approach for lineage specifier prediction in stem cell subpopulations. *NPJ Syst. Biol. Appl.*, **1**, 15012.

Ollion, J., Cochennec, J., Loll, F., Escudé, C., and Boudier, T. (2013). TANGO: a generic tool for high-throughput 3D image analysis for studying nuclear organization. *Bioinformatics*, **29**(14), 1840–1841.

Omranian, N., Eloundou-Mbebi, J. M. O., Mueller-Roeber, B., and Nikoloski, Z. (2016). Gene regulatory network inference using fused LASSO on multiple data sets. *Sci. Rep.*, **6**, 20533.

Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., Campbell, N. H., Chavali, G., Chen, C., del Toro, N., Duesbury, M., Dumousseau,

M., Galeota, E., Hinz, U., Iannuccelli, M., Jagannathan, S., Jimenez, R., Khadake, J., Lagreid, A., Licata, L., Lovering, R. C., Meldal, B., Melidoni, A. N., Milagros, M., Peluso, D., Perfetto, L., Porras, P., Raghunath, A., Ricard-Blum, S., Roechert, B., Stutz, A., Tognolli, M., van Roey, K., Cesareni, G., and Hermjakob, H. (2013). The MIntAct project—IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res.*, **42**(D1), D358–D363.

Ortega, J. M. and Rheinboldt, W. C. (1970). *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press.

Osborne, M. R., Presnell, B., and Turlach, B. A. (2000). On the LASSO and its dual. *J. Comp. Graph. Stat. (DEPRECATED, USE j-comp-graph-stat INSTEAD)*, **9**(2), 319–337.

Oshlack, A., Robinson, M. D., and Young, M. D. (2010). From RNA-seq reads to differential expression results. *Genome Biol.*, **11**(12), 1–10.

Pal, R., Datta, A., Bittner, M. L., and Dougherty, E. R. (2004). Intervention in context-sensitive probabilistic boolean networks. *Bioinformatics*, **21**(7), 1211–1218.

Palmer, J. A., Wipf, D. P., Kreutz-Delgado, K., and Rao, B. D. (2005). Variational EM algorithms for non-Gaussian latent variable models. In *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, pages 1059–1066.

Parikh, J. R., Xia, Y., and Marto, J. A. (2012). Multi-edge gene set networks reveal novel insights into global relationships between biological themes. *PLoS One*, **7**(9), e45211.

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Found. Trends Optim.*, **1**(3), 123–231.

Parisi, G. (1998). *Statistical Field Theory*. Addison Wesley.

Park, T., Yi, S.-G., Kang, S.-H., Lee, S. Y., Lee, Y.-S., and Simon, R. (2003). Evaluation of normalization methods for microarray data. *BMC Bioinformatics*, **4**(1), 1–13.

Paul, G., Cardinale, J., and Sbalzarini, I. F. (2013). Coupling image restoration and segmentation: A generalized linear model/Bregman perspective. *Int. J. Comput. Vis.*, **104**(1), 69–93.

Pe'er, D., Regev, A., Elidan, G., and Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, **17**(Suppl 1), S215–S224.

Pelleg, D. and Moore, A. (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In *Proc. Int. Conf. Mach. Learn.*, pages 727–734, Stanford, CA, USA.

Pereyra, M. and McLaughlin, S. (2017). Fast unsupervised Bayesian image segmentation with adaptive spatial regularisation. *IEEE Trans. Image Process.*, **26**.

Pereyra, M., Dobigeon, N., Batatia, H., and Tourneret, J. (2012). Segmentation of skin lesions in 2-D and 3-D ultrasound images using a spatially coherent generalized rayleigh mixture model. *IEEE Trans. Med. Imag.*, **31**(8), 1509–1520.

Pereyra, M., Dobigeon, N., Batatia, H., and Tourneret, J. (2013). Estimating the granularity coefficient of a Potts-Markov random field within a Markov chain Monte Carlo algorithm. *IEEE Trans. Image Process.*, **22**(6), 2385–2397.

Perrin, B.-E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., and d'Alché Buc, F. (2003). Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, **19**(suppl 2), ii138–ii148.

Peyré, G. (2011). A review of adaptive image representations. *IEEE J. Sel. Topics Signal Process.*, **5**(5), 896–911.

Pham, M. Q., Duval, L., Chaux, C., and Pesquet, J.-C. (2014). A primal-dual proximal algorithm for sparse template-based adaptive filtering: Application to seismic multiple removal. *IEEE Trans. Signal Process.*, **62**(16), 4256–4269.

Pirayre, A., Couprie, C., Bidard, F., Duval, L., and Pesquet, J.-C. (2015a). BRANE Cut: biologically-related a priori network enhancement with graph cuts for gene regulatory network inference. *BMC Bioinformatics*, **16**(1), 369.

Pirayre, A., Couprie, C., Duval, L., and Pesquet, J.-C. (2015b). Fast convex optimization for connectivity enforcement in gene regulatory network inference. In *Proc. Int. Conf. Acoust. Speech Signal Process.*, pages 1002–1006, South Brisbane, QLD, Australia.

Pirayre, A., Couprie, C., Duval, L., and Pesquet, J.-C. (2015c). Graph inference enhancement with clustering: Application to gene regulatory network reconstruction. In *Proc. Eur. Sig. Image Proc. Conf.*, pages 2406–2410, Nice, France.

Pirayre, A., Zheng, Y., Pesquet, J.-C., and Duval, L. (2017). HOGMep: variational Bayes and higher-order graphical models applied to joint image recovery and segmentation. In *Proc. Int. Conf. Image Process.*, Beijing, China.

Pirayre, A., Couprie, C., Duval, L., and Pesquet, J.-C. (2018a). BRANE Clust: cluster-assisted gene regulatory network inference refinement. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **15**(3), 850–860.

Pirayre, A., Ivanoff, D., Duval, L., Blugeon, C., Firmo, C., Perrin, S., Jourdier, E., Margeot, A., and Bidard, F. (2018b). Growing *Trichoderma reseei* on a mix of carbon sources reveals links between development and cellulase production. *To be submitted (BMC Genomics)*.

Pirim, H., Ekşioğlu, B., Perkins, A. D., and Yüceer, Ç. (2012). Clustering of high throughput gene expression data. *Comput. Oper. Res.*, **39**, 3046–3061.

Pižurica, A., Zlokolica, V., and Philips, W. (2004). Noise reduction in video sequences using wavelet-domain and temporal filtering. In F. Truchetet, editor, *Proc. SPIE, Wavelet Appl. Indust. Process.*

Poggi-Parodi, D., Bidard, F., Pirayre, A., Portnoy, T., Blugeon, C., Seiboth, B., Kubicek, C. P., Le Crom, S., and Margeot, A. (2014). Kinetic transcriptome analysis reveals an essentially intact induction system in a cellulase hyper-producer *Trichoderma reesei* strain. *Biotechnol. Biofuels*, **7**(1).

Polynikis, A., Hogan, S., and di Bernardo, M. (2009). Comparing different ODE modelling approaches for gene regulatory networks. *J. Theor. Biol.*, **261**(4), 511–530.

Portilla, J., Strela, V., Wainwright, M., and Simoncelli, E. (2003). Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Process.*, **12**(11), 1338–1351.

Portnoy, T. (2011). *Analyse du transcriptome de* Trichoderma reesei *pour l'amélioration de la production de cellulases*. Ph.D. thesis, Université Pierre et Marie Curie.

Portnoy, T., Margeot, A., Linke, R., Atanasova, L., Fekete, E., Sándor, E., Hartl, L., Karaffa, L., Druzhinina, I., Seiboth, B., Le Crom, S., and Kubicek, C. (2011). The CRE1 carbon catabolite repressor of the fungus *Trichoderma reesei*: a master regulator of carbon assimilation. *BMC Genom.*, **12**, 269.

Prelich, G. (2012). Gene overexpression: Uses, mechanisms, and interpretation. *Genetics*, **190**(3), 841–854.

Prill, R. J., Marbach, D., Saez-Rodriguez, J., Sorger, P. K., Alexopoulos, L. G., Xue, X., Clarke, N. D., Altan-Bonnet, G., and Stolovitzky, G. (2010). Towards a rigorous assessment of systems biology models: the DREAM3 challenges. *PLoS One*, **5**(2), e9202.

Pustelnik, N., Benazza-Benhayia, A., Zheng, Y., and Pesquet, J.-C. (2016). Wavelet-based image deconvolution and reconstruction. In *Wiley Encyclopedia of Electrical and Electronics Engineering*. Wiley.

Qin, Y., Bao, L., Gao, M., Chen, M., Lei, Y., Liu, G., and Qu, Y. (2013). *Penicillium decumbens* BrlA extensively regulates secondary metabolism and functionally associates with the expression of cellulase genes. *Appl. Microbiol. Biotechnol.*, **97**(24), 10453–10467.

Quackenbush, J. (2002). Microarray data normalization and transformation. *Nat. Genet.*, **32**, 496–501.

Quang, D. and Xie, X. (2014). EXTREME: an online EM algorithm for motif discovery. *Bioinformatics*, **30**(12), 1667–1673.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.*, **66**(336), 846–850.

Rapaport, F., Zinovyev, A., Dutreix, M., Barillot, E., and Vert, J.-P. (2007). Classification of microarray data using gene networks. *BMC Bioinformatics*, **8**(1), 35.

Repetti, A., Pham, M. Q., Duval, L., Chouzenoux, E., and Pesquet, J.-C. (2015). Euclid in a taxicab: Sparse blind deconvolution with smoothed ℓ_1/ℓ_2 regularization. *IEEE Signal Process. Lett.*, **22**(5), 539–543.

Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., and Sabeti, P. C. (2011). Detecting novel associations in large data sets. *Science*, **334**(6062), 1518–1524.

Ressom, H. W., Zhang, Y., Xuan, J., Wang, Y., and Clarke, R. (2006). Inference of gene regulatory networks from time course gene expression data using neural networks and swarm intelligence. In *Proc. IEEE Symp. Comput. Intell. Bioinform. Computat. Biol.*, Toronto, Ontario, Canada.

Reymond, N. (2004). *Bioinformatique des puces à ADN et application à l'analyse du transcriptome de* Buchnera aphidicola. Ph.D. thesis, INSA Lyon.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer.

Robinson, M. D. and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.*, **11**(3), R25.

Robinson, M. D. and Smyth, G. K. (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, **23**(21), 2881–2887.

Robinson, M. D. and Smyth, G. K. (2008). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*, **9**(2), 321–332.

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2009). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**(1), 139–140.

Rodet, T., Orieux, F., Giovannelli, J.-F., and Abergel, A. (2008). Data inversion for over-resolved spectral imaging in astronomy. *IEEE J. Sel. Topics Signal Process.*, **2**(5), 802–811.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, **20**(0), 53 –65.

Roy, S., Lagree, S., Hou, Z., Thomson, J. A., Stewart, R., and Gasch, A. P. (2013). Integrated module and gene-specific regulatory inference implicates upstream signaling networks. *PLoS Comput. Biol.*, **9**(10), e1003252.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D*, **60**(1-4), 259–268.

Ruyssinck, J., Huynh-Thu, V. A., Geurts, P., Dhaene, T., Demeester, P., and Saeys, Y. (2014). NIMEFI: Gene regulatory network inference using multiple ensemble feature importance algorithms. *PLoS One*, **9**(3), e92709.

Saadatpour, A. and Albert, R. (2013). Boolean modeling of biological regulatory networks: A methodology tutorial. *Methods*, **62**(1), 3–12.

Saloheimo, A. (2000). Isolation of the *ace1* gene encoding a Cys2-His2 transcription factor involved in regulation of activity of the cellulase promoter *cbh1* of *Trichoderma reesei*. *J. Biol. Chem.*, **275**(8), 5817–5825.

Salwinski, L., Miller, C. S., Smith, A. J., Pettit, F. K., Bowie, J. U., and Eisenberg, D. (2004). The database of interacting proteins: 2004 update. *Nucleic Acids Res.*, **32**(90001), D449–D451.

Sandelin, A., Alkema, W., Engström, P., Wasserman, W. W., and Lenhard, B. (2004). JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.*, **32**(90001), 91–94.

Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statist. Appl. Genet. Mol.*, **4**(1).

Schaffter, T., Marbach, D., and Floreano, D. (2011). GeneNetWeaver: *in silico* benchmark generation and performance profiling of network inference methods. *Bioinformatics*, **27**(16), 2263–2270.

Scherer, A., editor (2009). *Batch Effects and Noise in Microarray Experiments: Sources and Solutions*. Wiley.

Schreiber, S. L. (2005). Small molecules: the missing link in the central dogma. *Nat. Chem. Biol.*, **1**(2), 64–66.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, **6**(2), 461–464.

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N. (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.*, **34**(2), 166–176.

Seiboth, B., Karimi, R. A., Phatale, P. A., Linke, R., Hartl, L., Sauer, D. G., Smith, K. M., Baker, S. E., Freitag, M., and Kubicek, C. P. (2012). The putative protein methyltransferase LAE1 controls cellulase gene expression in *Trichoderma reesei*. *Mol. Microbiol.*, **84**(6), 1150–1164.

Seidl, V., Gamauf, C., Druzhinina, I. S., Seiboth, B., Hartl, L., and Kubicek, C. P. (2008). The *Hypocrea jecorina* (*Trichoderma reesei*) hypercellulolytic mutant RUT c30 lacks a 85 kb (29 gene-encoding) region of the wild-type genome. *BMC Genom.*, **9**(1), 327.

Servant, N., Varoquaux, N., Lajoie, B. R., Viara, E., Chen, C.-J., Vert, J.-P., Heard, E., Dekker, J., and Barillot, E. (2015). HiC-pro: an optimized and flexible pipeline for Hi-C data processing. *Genome Biol.*, **16**(1).

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**(8), 888–905.

Shimamura, T., Imoto, S., Yamaguchi, R., Nagasaki, M., and Miyano, S. (2010). Inferring dynamic gene networks under varying conditions for transcriptomic network comparison. *Bioinformatics*, **26**(8), 1064–1072.

Shmulevich, I., Dougherty, E. R., Kim, S., and Zhang, W. (2002). Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, **18**(2), 261–274.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, **30**(3), 83–98.

Siddiqui, K. I., Hero, A. O., and Siddiqui, M. M. (2002). Mathematical morphology applied to spot segmentation and quantification of gene microarray images. In *Proc. Asilomar Conf. Signal Syst. Comput.*, pages 926–930, Pacific Grove, CA, USA.

Siegenthaler, C. and Gunawan, R. (2014). Assessment of network inference methods: How to cope with an underdetermined problem. *PLoS One*, **9**(3), e90481.

Silvescu, A. and Honavar, V. (2001). Temporal boolean network models of genetic networks and their inference from gene expression time series. *Complex systems*, **13**, 2001.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group lasso. *J. Comp. Graph. Stat. (DEPRECATED, USE j-comp-graph-stat INSTEAD)*, **22**(2), 231–245.

Singaraju, D., Grady, L., Sinop, A. K., and Vidal, R. (2011). Continuous valued MRFs for image segmentation. In A. Blake, P. Kohli, and C. Rother, editors, *Markov Random Fields for Vision and Image Processing*, pages 127–142. MIT Press.

Singh, N. and Vidyasagar, M. (2016). bLARS: An algorithm to infer gene regulatory networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **13**(2), 301–314.

Šmídl, V. and Quinn, A. (2006). *The Variational Bayes Method in Signal Processing*. Springer.

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Stat. Appl. Genet. Mol. Biol.*, **3**(1), 1–25.

Smyth, G. K. (2005). limma: linear models for microarray data. In R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, and W. Huber, editors, *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pages 397–420. Springer.

Smyth, G. K. and Speed, T. (2003). Normalization of cDNA microarray data. *Methods*, **31**(4), 265–273.

Sodjo, J., Giremus, A., Caron, F., Giovannelli, J.-F., and Dobigeon, N. (2016). Joint segmentation of multiple images with shared classes: a Bayesian nonparametrics approach. In *Proc. IEEE Workshop Stat. Signal Process.*, Palma de Majorca, Spain.

Soinov, L. A., Krestyaninova, M. A., and Brazma, A. (2003). Towards reconstruction of gene networks from expression data by supervised learning. *Genome Biol.*, **4**(1), R6.

Soneson, C. and Delorenzi, M. (2013). A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics*, **14**(1), 91.

Sonka, M. and Fitzpatrick, J. M., editors (2000). *Handbook of Medical Imaging*, volume Volume 2. Medical Image Processing and Analysis. SPIE Press.

Soranzo, N., Bianconi, G., and Altafini, C. (2007). Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: synthetic versus real data. *Bioinformatics*, **23**(13), 1640–1647.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, **9**(12), 3273–3297.

Stearns, F. W. (2010). One hundred years of pleiotropy: A retrospective. *Genetics*, **186**(3), 767–773.

Steinhaus, H. (1956). Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci.*, **Cl. III — Vol. IV**(12), 801–804.

Strauss, J., Mach, R. L., Zeilinger, S., Hartler, G., Stöffler, G., Wolschek, M., and Kubicek, C. P. (1995). Crel, the carbon catabolite repressor protein from *Trichoderma reesei*. *FEBS Lett.*, **376**(1-2), 103–107.

Stricker, A. R., Grosstessner-Hain, K., Würleitner, E., and Mach, R. L. (2006). Xyr1 (xylanase regulator 1) regulates both the hydrolytic enzyme system and D-xylose metabolism in *Hypocrea jecorina*. *Eukaryot. Cell*, **5**(12), 2128–2137.

Stricker, A. R., Mach, R. L., and de Graaff, L. H. (2008). Regulation of transcription of cellulases- and hemicellulases-encoding genes in *Aspergillus niger* and *Hypocrea jecorina* (*Trichoderma reesei*). *Appl. Microbiol. Biotechnol.*, **78**(2), 211–220.

Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *Science*, **302**(5643), 249–255.

Sugiyama, M., Azencott, C.-A., Grimm, D., Kawahara, Y., and Borgwardt, K. M. (2014). Multi-task feature selection on multiple networks via maximum flows. In *Proc. SIAM Int. Conf. Data Mining*, pages 199–207, Philadelphia, PA, USA.

Tamada, Y., Kim, S., Bannai, H., Imoto, S., Tashiro, K., Kuhara, S., and Miyano, S. (2003). Estimating gene networks from gene expression data by combining bayesian network model with promoter element detection. *Bioinformatics*, **19**(Suppl 2), ii227–ii236.

Tarabalka, Y., Chanussot, J., and Benediktsson, J. A. (2010). Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recogn.*, **43**(7), 2367–2379.

Taroni, J. N. and Greene, C. S. (2017). Cross-platform normalization enables machine learning model training on microarray and RNA-seq data simultaneously. *PREPRINT*.

Thomas-Cholier, M., Sand, O., Turatsinze, J., Janky, R., Defrance, M., Vervish, E., Brohee, S., and van Helden, J. (2008). RSAT: regulatory sequence analysis tools. *Nucleic Acids Res.*, **36**(Web Server issue), W119–W127.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **58**(1), 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **67**(1), 91–108.

Tikhonov, A. N. (1963). On the solution of ill-posed problems and the method of regularization. *Dokl. Akad. Nauk SSSR*, **151**, 501–504.

Toh, H. and Horimoto, K. (2002). Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics*, **18**(2), 287–297.

Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Nat. Acad. Sci. U.S.A.*, **98**(9), 5116–5121.

Unger, M., Pock, T., Trobin, W., Cremers, D., and Bischof, H. (2008). TVSeg - interactive total variation based image segmentation. In *Proc. Brit. Machine Vis. Conf.*, pages 40.1–40.10, Leeds, UK. British Machine Vision Association and Society for Pattern Recognition.

Van De Ville, D., Demesmaeker, R., and Preti, M. G. (2017). When Slepian meets Fiedler: Putting a focus on the graph spectrum. *IEEE Signal Process. Lett.*

van Someren, E. P., Vaes, B. L. T., Steegenga, W. T., Sijbers, A. M., Dechering, K. J., and Reinders, M. J. T. (2005). Least absolute regression network analysis of the murine osteoblast differentiation network. *Bioinformatics*, **22**(4), 477–484.

Vert, J.-P. (2013). Les applications industrielles de la bio-informatique. *Annales des Mines — Réalités industrielles*, pages 17–23.

Vignes, M., Vandel, J., Allouche, D., Ramadan-Alban, N., Cierco-Ayrolles, C., Schiex, T., Mangin, B., and de Givry, S. (2011). Gene regulatory network reconstruction using Bayesian networks, the Dantzig selector, the lasso and their meta-analysis. *PLoS One*, **6**(12), e29165.

Vinh, N. X., Chetty, M., Coppel, R., and Wangikar, P. P. (2012). Gene regulatory network modeling via global optimization of high-order dynamic Bayesian network. *BMC Bioinformatics*, **13**(1), 131.

von Mering, C., Jensen, L. J., Snel, B., Hooper, S. D., Krupp, M., Foglierini, M., Jouffre, N., Huynen, M. A., and Bork, P. (2005). STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Res.*, **33 (Suppl. 1)**, D433–D437.

Wahde, M. and Hertz, J. (2000). Coarse-grained reverse engineering of genetic regulatory networks. *BioSystems*, **55**(1-3), 129–136.

Wand, M., Ormerod, J. T., Padoan, S. A., and Fruhwirth, R. (2010). Variational Bayes for elaborate distributions. *Centre for Statistical and Survey Methodology, University of Wollongong, Working Paper*.

Wang, J., Ma, J. Z., and Li, M. D. (2004). Normalization of cDNA microarray data using wavelet regressions. *Comb. Chem. High Throughput Screen.*, **7**(8), 783–791.

Wang, M., Jiang, N., Jia, T., Leach, L., Cockram, J., Waugh, R., Ramsay, L., Thomas, B., and Luo, Z. (2012a). Genome-wide association mapping of agronomic and morphologic traits in highly structured populations of barley cultivars. *Theor. Appl. Genet.*, **124**(2), 233–246.

Wang, R.-S., Saadatpour, A., and Albert, R. (2012b). Boolean modeling in systems biology: an overview of methodology and applications. *Phys. Biol.*, **9**(5), 055001.

Wang, Y., Joshi, T., Zhang, X.-S., Xu, D., and Chen, L. (2006). Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics*, **22**(19), 2413–2420.

Wasserstein, R. L. and Lazar, N. A. (2016). The ASA's statement on p-values: context, process, and purpose. *Am. Stat.*, **70**(2), 129–133.

Weaver, D. C., Workman, C. T., and Stormo, G. D. (1999). Modeling regulatory networks with weight matrices. In *Pac. Symp. Biocomput.*

Werhli, A. V. and Husmeier, D. (2007). Reconstructing gene regulatory networks with bayesian networks by combining expression data with multiple sources of prior knowledge. *Stat. Appl. Genet. Mol. Biol.*, **6**(1).

Whittaker, E. T. (1922). On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society*, **41**, 63–75.

Whittaker, J. (1990). *Graphical models in applied multivariate statistics*. Probability and mathematical statistics. Wiley.

Wille, A., Zimmermann, P., Vranoá, E., Füholz, A., Laule, O., Bleuler, S., Hennig, L., Prelić, A., von Rohr, P., Thiele, L., Zitzler, E., Gruissem, W., and Bühlmann, P. (2004). Sparse graphical Gaussian modeling of the isoprenoid gene network in *Arabidopsis thaliana*. *Genome Biol.*, **5**(11), R92.

Workman, C., Jensen, L. J., Jarmer, H., Berka, R., Gautier, L., Nielser, H. B., Saxild, H.-H., Nielsen, C., Brunak, S., and Knudsen, S. (2002). A new non-linear normalization method for reducing variability in DNA microarray experiments. *Genome Biol.*, **3**(9), 0048.1–0048.16.

Wu, S., Liu, Z.-P., Qiu, X., and Wu, H. (2014). Modeling genome-wide dynamic regulatory network in mouse lungs with influenza infection using high-dimensional ordinary differential equations. *PLoS One*, **9**(5), e95276.

Xu, R., Venayagamoorthy, G. K., and Wunsch, D. C. (2007). Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. *Neural Netw.*, **20**(8), 917–927.

Yang, I. V., Chen, E., Hasseman, J. P., Liang, W., Frank, B. C., Wang, S., Sharov, V., Saeed, A. I., White, J., Li, J., Lee, N. H., J., Y. T., and J., Q. (2002a). Within the fold: assessing differential expression measures and reproducibility in microarray assays. *Genome Biol.*, **3**(11), research0062.1–research0062.12.

Yang, Y. H., Dudoit, S., Luu, P., and Speed, T. P. (2001). Normalization for cDNA microarry data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty, editors, *Proc. SPIE*, volume 4266 of *Microarrays: Optical Technologies and Informatics*, pages 141–152.

Yang, Y. H., Dudoit, S., Luu, P., Lin, D. M., Peng, V., Ngai, J., and Speed, T. P. (2002b). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.*, **30**(4), e15.

Yanming, D., Schafer, D. W., Cumbie, J. S., and Chang, J. H. (2011). The NBP negative binomial model for assessing differential gene expression from RNA-Seq. *Stat. Appl. Genet. Mol. Biol.*, **10**(1), 1–28.

Yeung, M. K. S., Tegner, J., and Collins, J. J. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proc. Nat. Acad. Sci. U.S.A.*, **99**(9), 6163–6168.

Young, W., Raftery, A. E., and Yeung, K. (2014). Fast Bayesian inference for gene regulatory networks using ScanBMA. *BMC Syst. Biol.*, **8**(1), 47.

Yu, F. Y. (1982). The Potts model. *Rev. Mod. Phys.*, **54**(1).

Yu, J., Smith, V. A., Wang, P. P., Hartemink, A. J., and Jarvis, E. D. (2004). Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, **20**(18), 3594–3603.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **68**(1), 49–67.

Zhang, A., Tang, C., and Jiang, D. (2004). Cluster analysis for gene expression data: A survey. *IEEE Trans. Knowl. Data Eng.*, **16**(11), 1370–1386.

Zhang, X., Liu, K., Liu, Z.-P., Duval, B., Richer, J.-M., Zhao, X.-M., Hao, J.-K., and Chen, L. (2013). NARROMI: a noise and redundancy reduction technique improves accuracy of gene regulatory network inference. *Bioinformatics*, **29**(1), 106–113.

Zhao, N., Basarab, A., Kouame, D., and Tourneret, J.-Y. (2016a). Joint segmentation and deconvolution of ultrasound images using a hierarchical bayesian model based on generalized gaussian priors. *IEEE Trans. Image Process.*, **25**(8), 3736–3750.

Zhao, S., Yan, Y.-S., He, Q.-P., Yang, L., Yin, X., Li, C.-X., Mao, L.-C., Liao, L.-S., Huang, J.-Q., Xie, S.-B., Nong, Q.-D., Zhang, Z., Jing, L., Xiong, Y.-R., Duan, C.-J., Liu, J.-L., and Feng, J.-X. (2016b). Comparative genomic, transcriptomic and secretomic profiling of *Penicillium oxalicum* HP7-1 and its cellulase and xylanase hyper-producing mutant EU2106, and identification of two novel regulatory genes of cellulase and xylanase gene expression. *Biotechnol. Biofuels*, **9**(1).

Zhao, W., Serpedin, E., and Dougherty, E. R. (2008). Inferring connectivity of genetic regulatory networks using information-theoretic criteria. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **5**(2), 262–274.

Zheleva, E., Getoor, L., and Sarawagi, S. (2010). Higher-order graphical models for classification in social and affiliation networks. In *NIPS Workshop on Networks Across Disciplines: Theory and Applications*.

Zheng, Y., Fraysse, A., and Rodet, T. (2015a). Efficient variational Bayesian approximation method based on subspace optimization. *IEEE Trans. Image Process.*, **24**(2), 681–693.

Zheng, Y., Fraysse, A., and Rodet, T. (2015b). Wavelet based unsupervised variational Bayesian image reconstruction approach. In *Proc. Eur. Sig. Image Proc. Conf.*, Nice, France.

Zibulevsky, M. and Pearlmutter, B. A. (2001). Blind source separation by sparse decomposition in a signal dictionary. *Neural Comput.*, **13**(4), 863–882.

Zien, A., Aigner, T., Zimmer, R., and Lengauer, T. (2001). Centralization: a new method for the normalization of gene expression data. *Bioinformatics*, **17**(Suppl. 1), S323–S331.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **67**(2), 301–320.