Modélisation de phénomènes biologiques complexes : application à l'étude de la réponse antigénique de lymphocytes B sains et tumoraux

Nicolas Jung

UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE DES SCIENCES DE LA VIE ET DE LA SANTE

ImmunoRhumatologie Moléculaire, INSERM UMR_S 1109

THÈSE présentée par :

Nicolas JUNG

soutenue le : **3 décembre 2014**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Biologie des systèmes

Modélisation de phénomènes biologiques complexes : application à l'étude de la réponse antigénique de lymphocytes B sains et tumoraux.

THÈSE dirigée par :

Pr Seiamak BAHRAM	Université de Strasbourg

RAPPORTEURS :

DR Charles AUFFRAY	CNRS
DR Yann GUERMEUR	CNRS

AUTRES MEMBRES DU JURY :

Pr Philippe GEORGEL	Université de Strasbourg
Dr Myriam MAUMY-BERTRAND	Université de Strasbourg
Pr Rodolphe THIEBAUT	Université de Bordeaux

Jury

Le jury officiel présidant à la soutenance de cette thèse est composé de :

Seiamak Bahram, directeur de thèse

Charles Auffray, rapporteur externe

Yann Guermeur, rapporteur externe

Philippe Georgel, rapporteur interne

Myriam Maumy-Bertrand, examinatrice et co-encadrante

Rodolphe Thiébaut, examinateur

Ce jury est complété par la présence, en qualité d'invité, des personnes suivantes :

Frédéric Bertrand, membre invité

Laurent Vallat, membre invité

Remerciements

Je dirai hautement : Voilà ce que j'ai fait, ce que j'ai pensé, ce que je fus. J'ai dit le bien et le mal avec la même franchise. Je n'ai rien tu de mauvais, rien ajouté de bon, et s'il m'est arrivé d'employer quelque ornement indifférent, ce n'a jamais été que pour remplir un vide occasionné par mon défaut de mémoire ; j'ai pu supposer vrai ce que je savais avoir pu l'être, jamais ce que je savais être faux.

Préambule des Confessions de Jean-Jacques Rousseau

L'écriture des remerciements est un moment privilégié où l'on se permet de laisser porter son regard vers les jours qui ont passé avec l'humilité de celui qui sait que s'il avait été seul tant de choses n'auraient pas été possibles - ou auraient été bien plus compliquées à réaliser. Il s'agit là de reconnaître que la réussite, quelle qu'elle soit, n'est jamais entièrement personnelle, mais doit être considérée à la fois comme l'aboutissement de longs efforts, d'un rude travail, de sacrifices et de la concrétisation de l'héritage et de la bienveillance de personnes qu'il s'agit de mettre à l'honneur ici. Cette thèse aura été ponctuée d'instants de diverses fortunes, mais l'essentiel est que je ne me suis jamais retrouvé seul face à mes inquiétudes et mes questionnements. L'écriture des remerciements est donc aussi une démarche cathartique dont l'aboutissement est l'effacement des moments de désenchantement au profit de l'évocation de toute l'empathie dont mon entreprise a fait l'objet. Il est à la fois amusant et effrayant de se rendre subitement compte que l'on aurait pu vivre mille fois la même vie et ne jamais parvenir à un tel accomplissement si la providence n'avait, de ci, de là, intercédé en notre faveur. Et, qu'est-ce que la providence, sinon des mains tendues et des mots d'encouragement ? J'essayerai de n'oublier personne et de trouver une place pour tous ceux qui peuvent se reconnaître dans les mots que je viens d'écrire. C'est une tâche qui est difficile, et sans doute, quelques jours après ma soutenance, surgira devant moi le visage, le sourire de celui ou celle que j'aurai oublié. Je leur présente d'avance mes excuses.

Je remercie dans un premier temps le Professeur Seiamak Bahram, qui, en tant que directeur de thèse, a su donner à cette entreprise un cadre et des fondations sur lesquelles j'ai pu m'appuyer dans mon travail. Je le remercie

aussi d'avoir fait le pari de l'interdisciplinarité, de m'avoir pris comme doctorant avec ma formation de statisticien. C'est un pari plus audacieux qu'il ne figure à première vue pour celui qui sait combien les mathématiques et la biologie sont des univers singulièrement différents. Les mots eux-mêmes travestissent leur sens d'une discipline à l'autre, comme ces faux amis dont on vous fait la liste dans les cours de langue. Les méthodes, aussi, sont différentes, et avec elles, les façons de réfléchir. J'ai beaucoup appris de cette confrontation et nul doute que les années à venir me montreront à quel point elle aura été profitable. Merci donc à lui.

Je remercie également Myriam Maumy-Bertrand dont le soutien aura été constant au cours de mes années de thèse. Elle a co-encadré cette thèse en étant toujours disponible et son énergie a été pour moi une source de motivation supplémentaire. Ses propositions, ses conseils, ses suggestions ont toujours aidé efficacement mon travail. Je tiens particulièrement à la remercier pour sa confiance et pour la patience dont elle a fait preuve. Malheureusement, il est arrivé à plusieurs moments que les difficultés rencontrées lors de ma thèse ne soient pas de nature scientifique ; cependant, elle aura toujours été là. C'est donc très chaleureusement que la remercie.

Je remercie ensuite le Directeur de Recherche Yann Guermeur qui a accepté de rapporter cette thèse et de faire partie de mon jury de mi-thèse. Je le remercie également pour l'ensemble des conseils qu'il a pu me donner, ainsi que pour les pistes qu'il m'a proposées. Son regard d'apprentissien sur mon travail a été très enrichissant.

Je remercie le Directeur de recherche Charles Auffray d'avoir accepté d'être rapporteur externe de cette thèse.

Je remercie le Professeur Philippe Georgel d'avoir accepté d'être rapporteur interne de cette thèse, ainsi que d'avoir fait partie de mon jury de mi-thèse. Il m'a donné des conseils très utiles, en particulier quant à la rédaction de ce manuscrit.

Je remercie également le Professeur Rodolphe Thiébaut d'avoir accepté de faire partie du jury de cette thèse.

Je remercie également Frédéric Bertrand avec qui j'ai travaillé sur de nombreux projets. Toujours disponible, toujours prêt à discuter et à vous proposer des idées pertinentes, il apparaît comme étant le statisticien, qui mélangeant curiosité, connaissance, imagination et rigueur fait de la statistique une discipline réellement passionnante. J'ai pris beaucoup de plaisir à travailler avec lui.

Je remercie le Docteur Laurent Vallat de m'avoir proposé de travailler sur ce sujet que j'ai trouvé passionnant. J'eusse simplement souhaité que le poids du secret permanent fut remplacé par une collaboration plus étroite.

Je remercie également toutes les personnes de mon équipe d'accueil avec qui j'ai pu travailler. Je pense d'abord à Raphaël Carapito avec lequel j'ai

passé beaucoup de temps et qui m'a toujours semblé curieux et intéressé par les idées de modélisations statistiques que j'ai pu lui proposer. Je pense ensuite à Nicodème Paul, Ghada Alsaleh, Barning Cindy avec qui j'ai également pu travailler. Je n'oublie évidemment pas les autres membres de l'équipe.

Je tiens cependant à remercier d'une façon particulière Ramzy, Pilar, Pierre, Wassila, Marion, et Cédric avec qui j'ai partagé des moments agréables et de détente.

Je tiens également à remercier tous les membres de la faculté de mathématiques avec qui j'ai pu interagir, que ce soit au niveau des cours de leur organisation ou de projets scientifiques. J'ai une pensée particulière pour Phillipe Helluy pour le soutien qu'il m'a apporté.

Évidemment, il me serait impossible de passer sous silence toutes les discussions, toutes les sorties, toutes les parties de foot, et toutes ces choses qu'il ne convient pas d'écrire dans un manuscrit de thèse mais agrémentent le temps et font oublier les déceptions. Alors merci à vous tous : Ambroise, Philippe, Auguste, Olivier, Aurélien, Jérémy... Je garde aussi une place toute spéciale pour mes anciens co-bureaux, Alexandre, Florian et Amaury.

Je n'oublie évidemment pas mon nouveau co-bureau et collègue Théo avec lequel nous continuerons à travailler dans la joie et la bonne humeur.

Merci également à tout le personnel administratif et technique sans lequel nous ne pourrions pas travailler. J'ai une pensée spéciale pour Delphine Schmitt et Laure Ziraoui.

Comme je l'écrivais plus haut, la thèse est un moment singulier : elle ferme l'âge d'or des études et ouvre la route vers le monde professionnel. Toute ma vie de travail se résume à cet instant comme un chemin, un long chemin parfois tortueux, vers cette soutenance. Après elle commenceront les expériences que j'espère heureuses et variées. La "vraie" vie, paraît-il. Mais durant ce chemin, je n'oublierai jamais à quel point j'ai toujours pu compter sur mes parents. Ils m'ont encouragé sans relâche dans toutes mes initiatives. Je veux qu'ils comprennent que je partage ma réussite avec eux. Je ne connais pas d'encouragement plus fort et plus sincère que ceux des parents pour leurs enfants.

Merci également à mon petit frère, Mathieu, qui longtemps a été mon compagnon d'aventures. Il a toujours été là pour moi, pour m'encourager et pour toutes ces choses parfois pleines de malice que l'on s'autorise entre frères. Merci mon Biberle !

Enfin, merci à celle qui partage ma vie, c'est-à-dire mes joies et mon bonheur mais aussi tous ces moments de doute, d'angoisse, de crispation. Elle m'a montré que la vie n'a de saveur que si elle est partagée, que l'in-

quiétude est moins prégnante quand elle est présente pour me réconforter. Je la remercie tout simplement pour tous les moments de bonheur qu'elle m'a fait vivre. Merci du fond du cœur, ma petite Marion.

Strasbourg, le 14 janvier 2015.

Table des matières

Table des matières **ix**

Liste des figures **xiii**

Préface **1**

I Introduction **5**

1 Biologie des systèmes complexes **7**
1.1 Systèmes complexes 8
1.2 Biologie des systèmes complexes 10
1.3 Réseaux de régulation génique 12
1.4 Généralités sur les réseaux 13
1.4.1 Définition d'un réseau 13
1.4.2 Représentation d'un réseau 14
1.4.3 Caractérisation d'un réseau 15
1.4.4 Caractérisation d'un nœud dans le réseau 17
1.4.5 Quelques topologies classiques de réseaux 17
1.5 Inférence de réseaux de régulation génique 21
1.5.1 Algorithmes et méthodes 23
1.6 Quelle utilité pour les réseaux biologiques 28
1.7 De la biologie des systèmes à la médecine des systèmes 31
1.8 Objectifs de cette thèse 32

2 Régression Lasso **35**
2.1 Introduction et historique 35
2.2 Premières propriétés de la régression Lasso 39
2.3 Pourquoi la norme $\mathcal{L}_1$? 42
2.4 Conditions de Karush-Kuhn-Tucker 43
2.5 L'algorithme LARS-Lasso 44
2.6 Résultats théoriques pour le Lasso 50
2.6.1 Un peu de formalisme 50
2.6.2 Cas où N>M 52
2.6.3 Cas $M \geqslant N$ 53
2.6.4 Limites théoriques de la sélection de variables 55
2.7 Alternatives et améliorations du Lasso 56
2.7.1 Réduire le biais 57
2.7.2 Lasso et corrélation 57
2.7.3 Tests pour le Lasso 59
2.7.4 Exemple 60

2.8 Conclusion 62

3 Modèle biologique **65**
3.1 La leucémie lymphoïde chronique comme modèle biologique 65
3.2 Plan d'expérience de l'étude 66
3.2.1 Puces à ADN 67
3.2.2 Analyse succincte du jeu de données 68
3.3 Conclusion 73

II Vers une modulation orientée dans un programme génique **77**

4 Modélisation en cascade d'un réseau de gène **79**
4.1 Modélisation en cascade 79
4.1.1 Clustering 80
4.1.2 Modèle linéaire pour l'inférence du réseau 89
4.1.3 Conclusion 94
4.2 Modifications du modèle initial 95
4.2.1 Etape de clustering 95
4.2.2 Estimation des matrices $F_{m(i)m(j)}$ 98
4.2.3 Estimation du réseau 100
4.3 Comparaison de notre méthode d'inférence de réseau 101
4.3.1 Méthodes pour la comparaison 101
4.3.2 Comparaison sur notre jeu de données 102
4.3.3 Comparaison *in silico* 102
4.4 Conclusion 103

5 Reverse-engineering **105**
5.1 Abstract 105
5.2 Introduction 106
5.3 Results 108
5.3.1 Gene selection and network reverse-engineering 108
5.3.2 Application to synthetic data 108
5.3.3 Application to the CLL data set 110
5.4 Discussion 114
5.5 Materials and methods 115
5.5.1 Gene selection 115
5.5.2 Model inference 116
5.5.3 Simulations 117
5.5.4 Microarrays, RNA interference and validation experiments 118
5.6 Acknoledgments 119

6 Cascade **121**
6.1 Abstract 122
6.2 Introduction 122
6.3 Details on the package features 122
6.3.1 Gene selection and cluster assignment 123
6.3.2 Reverse-engineering of the network 123
6.3.3 Prediction 124

6.3.4 Simulation 124
6.4 Examples 124
6.5 Figures 124

7 Compléments sur le package Cascade **127**
7.1 Sélection des gènes 127
7.2 Inférence du réseau 128
7.2.1 Choix du seuil 128

8 selectBoost **133**
8.1 Abstract 133
8.2 Introduction 133
8.3 Methods 137
8.3.1 Introduction 137
8.3.2 The selectBoost algorithm 138
8.3.3 Choosing the parameters of the algorithm 139
8.4 Numerical studies 141
8.4.1 Introduction 141
8.4.2 Results of the simulation 142
8.5 Application to the diabetes dataset 145
8.6 Discussion 145

III Perspectives et conclusions 147

9 Vers une modulation orientée du programme génique **149**
9.1 Introduction 149
9.2 De Cascade vers Patterns 151
9.3 Le choix de la normalisation 151
9.4 Le choix de la méthode de sélection 152
9.5 Procédure suivie 153
9.6 Résultats obtenus 153

10 Conclusions **155**
10.1 Résultats 155
10.2 Perspectives de travail 156
10.3 Conclusion 157

IV Annexes 159

A Supporting information for Chapter 3 **161**

B Application 1 : **167**
B.1 Introduction 167
B.2 Installation requirements 169
B.3 Data pre-processing 169
B.4 Gene selection 171
B.5 Gene regulatory network reverse-engineering . . . 177
B.5.1 Theoretical background 177
B.5.2 Performing the reverse-engineering algorithm 179
B.5.3 Choosing the best cutoff for edge minimal strength 182

B.5.4 Analyzing the network 184
B.6 Prediction of gene expression modulations after a knock-out experiment 185
B.7 Simulation 185

C Application 2 **191**
C.1 Introduction 191
C.2 Importing the "E-MTAB-1475" dataset 191
C.3 Gene selections and reverse-engineering of the networks 192
C.3.1 Note on the "geneSelection" function 192
C.3.2 Selecting differentially expressed genes without specifying patterns 193
C.3.3 Selecting differentially expressed genes with specific patterns 196
C.3.4 Reverse-engineering the TH1 specific network 197
C.3.5 Using very specific patterns and lower the log-fold change threshold 201
C.3.6 Reverse-engineering an additional network 202
C.4 Prediction of gene expressions after a knock-out experiment 205

D Supplement information for Chapter 3 **207**
D.1 Confidence index 207
D.2 Comparisons 208
D.3 Example of results : modified BIC (BIC2) 211

E Réseaux multi-états **217**

Bibliographie **225**

Liste des figures

1 Notre processus de travail, qui s'inscrit parfaitement dans le cadre de la biologie des systèmes complexes, est le suivant : un problème biologique est posé dans un premier temps ; face à ce dernier, deux travaux s'engagent en parrallèle. D'une part, nous trouvons le travail du biologiste, dont le but est de créer un modèle biologique censé contraindre le problème biologique dans un cadre contrôlable, avant de procéder aux expériences nécessaires. D'autre part, le mathématicien a pour rôle de proposer un modèle statistique adapté au modèle biologique, avant de se servir de ce dernier pour prédire les résultats des interventions des expériences biologiques. Une comparaison doit ensuite être effectuée entre prédictions mathématiques et résultats d'expériences biologiques. Si ces résultats ne sont pas convaincants, il convient pour le biologiste d'affiner son modèle et de procéder à de nouvelles expériences, et au mathématicien d'affiner son modèle statistique et procéder aux nouvelles prédictions. Ce processus doit être répété jusqu'à ce que les comparaisons entre prédictions et résultats d'expérience coïncident. 2

1.1 Evolution du prix d'un séquençage complet d'un génome. . . 11
1.2 Systèmes complexes en biologie (Ideker *et al.* 2006). 12
1.3 Résumé du principe du réseau de régulation génique. 13
1.4 Représentation d'un réseau sans et avec direction (resp. figure de gauche et de droite). 14
1.5 Représentation d'un réseau avec des informations supplémentaires (nature du lien à gauche, nature du lien et indicateur de confiance à droite). 15
1.6 a : le degré sortant du nœud 1 est de 3. b : le degré entrant du nœud 1 est de 3. c : les degrés sortant et entrant du nœud 1 sont de 3. 15
1.7 Réseau aléatoire contenant 100 nœuds et 500 arêtes engendré selon le principe d'Erdös et Rényi. 19
1.8 Dans un réseau aléatoire à 1000 nœuds, évolution de la longueur moyenne du plus court chemin en fonction de la probabilité p de présence d'une arête dans le réseau. 19
1.9 D'une structure régulière à une structure aléatoire. À gauche : structure parfaitement régulière de 100 nœuds. Au milieu : chaque arête est modifiée aléatoirement avec une probabilité de $p = 0.1$. À droite : toutes les arêtes sont modifiées aléatoirement, $p = 1$. 20

1.10 D'une structure régulière à une structure aléatoire, analyse de la moyenne des plus courts chemins moyens et du cœfficient de clustering. 21
1.11 Exemple de réseau invariant d'échelle 22
1.12 Impacts des perturbations dans les systèmes biologiques. Figure issue de l'article Villaverde *et al.* (2013). 33

2.1 La fonction de pénalité du Lasso (à gauche) et celle du Ridge (à droite). 38
2.2 La fonction de pénalité dérivée du Lasso (à gauche) et celle du Ridge (à droite). 38
2.3 La boule unité $\mathcal{L}_1$, l'estimation Lasso obtenue à partir de l'estimation MCO. 40
2.4 La boule unité $\mathcal{L}_2$, l'estimation Ridge obtenue à partir de l'estimation MCO. 40
2.5 Exemple de fonctionnement de l'algorithme LARS-Lasso en deux dimensions. 49
2.6 Évolution de la cohérence mutuelle en fonction du nombre de variables, pour un nombre d'observations fixé à 25. 58
2.7 Evolution des cœfficients de régression en fonction du paramètre λ. A gauche le modèle 1 dans lequel les deux covariables du support se détachent nettement (rouge et noir). A droite le modèle deux, dans lequel la deuxième variable d'intérêt $\boldsymbol{x_{.2}}$ voit son cœfficient se détacher (rouge) tandis que la première variable d'intérêt (rose) apparait de façon alternative avec la covariable 10 (en bleue). 61
2.8 Evolution des cœfficients de régression en fonction du paramètre λ. La courbe violette représente les deux covariables correllées $\boldsymbol{x_{.1}}$ et $\boldsymbol{x_{.10}}$, la rouge la deuxième variable d'intérêt du modèle $\boldsymbol{x_{.2}}$ et en cyan une variable hors d'intérêt $\boldsymbol{x_{.4}}$. . . . 63
2.9 Evolution des cœfficients de régression en fonction du paramètre λ De haut en bas : $\boldsymbol{x_{.2}}$, puis, groupés $\boldsymbol{x_{.1}}$, $\boldsymbol{x_{.4}}$ et $\boldsymbol{x_{.10}}$ puis quelques variables hors d'intérêts. 63

3.1 Image venant d'une micropuce à ADN. Chaque point correspond à un probeset particulier. 67
3.2 Image venant d'une micropuce à ADN. Chaque point correspond à un probeset particulier. 68
3.3 Image venant d'une micropuce à ADN. Chaque point correspond à un probe-set particulier. 69
3.4 Classification des expressions de gènes retenus après sélection. Chaque graphique est divisé en trois parties. La partie de droite correspond aux expressions des individus sains, celle du milieu aux patients atteints de la forme indolente de la LLC et celle de gauche aux patients atteints de la forme agressive. 70
3.5 Les différents patterns temporels dans les expressions différentielles des gènes. 74
3.6 Illustration de la notion de réseau en cascade. 74

3.7 PLS-DA parcimonieuse : les expressions temporelles précoces ne permettent pas de distinguer les différents patients, contrairement aux expressions des temps tardifs. 75
3.8 PLS-DA parcimonieuse : les expressions précoces, analysées à part, permettent de discriminer les différents individus. . . 75
3.9 PLS-DA parcimonieuse : les expressions tardives, analysées à part, permettent de discriminer les différents individus. . . . 76

4.1 Exemple d'application de l'algorithme de clustering avec quatre gènes et deux temps de mesure (les répétitions liées aux patients ne sont pas représentées ici). 80
4.2 La queue de la loi exponentielle étant plus lourde que celle de loi normale, nous préférerons choisir la loi exponentielle pour des observations avec un grand niveau d'expression, tandis que le bruit sera retenu dans la loi normale. 96
4.3 La modélisation exponentielle favorise les observations contenant un outlier 96
4.4 Différentes lois de Laplace, avec p le paramètre de position, et s le paramètre de dispersion. 97

5.1 Results of gene selection. Representation of selected genes for a representative patient. Graphs (a) to (d) successively represent genes that have consistent up-regulation at a given time stressed in bold (t_1 to t_4 respectively). Graph (e) shows genes that are highly expressed through all four time points. Graph (f) shows all the retained genes. 112
5.2 Visualization of inferred networks. The gene regulatory network of the most aggressive leukemic B-cells (a), the indolent leukemic B-cells (b), and healthy B-cells (c) are represented. Nodes represent genes and edges statistical relationships between genes. For each network, hubs are highlighted in color. As the number of hubs decreases between aggressive, indolent and healthy networks, the structure of the network is changed. Bottom graphs represent sub-networks for DUSP1 (d) and EGR1 (e) in the most aggressive leukemic B-cells network. The concerned gene is highlighted in red. Direct links are represented in navy blue and indirect links are represented in pale blue. EGR1 is a gene whose influence is very large, since its subnetwork takes a large part of the complete network. In contrast, DUSP1 has a limited subnetwork. Visualization generated using R and R package Igraph. 113

6.1 Step 1 : gene selection in GSE39411 and assignment to a time cluster. 125
6.2 Step 2 : reverse-engineering of the network in GSE39411. Nodes represent genes and the arrows statistical links between the genes. Arrows' thickness depicts the intensity of the link. 125

6.3 Step 3 : predicted perturbations in the network, at the 2nd time point, after gene expression modulation at an early time in the temporal GRN of GSE39411. The green influential gene is supposed to be knocked-down. Color scale legend from downregulated (blue) to upregulated (red) genes. 125

7.1 Réseau sans seuillage : il y a une forte densité de liens, dont certains sont très proches de la nullité. 129
7.2 Réseau avec seuillage : la structure principale du réseau est apparente. 129
7.3 Choix du seuil en fonction de la valeur p du test d'adéquation à une distribution de type puissance. Les différentes zones de choix préférentiel permettent de choisir le seuil optimal parmi tous les seuils résultants en une valeur p supérieure à 0,1. . . 130

8.1 In this example $N = 20, P = 10, \boldsymbol{\beta} = (1, 1, 0, ..., 0)'$. The mean correlation between $\boldsymbol{x}_{1\cdot}$ and the other variables is 0.20 while the mean correlation between all the other variables is 0.95. The x-axis corresponds to the value of the penalty parameter λ ; the greater the parameter, the stronger the constraint. Left : with the lasso regression, no regularization is made. Right : with the elastic net regression, the coefficients of correlated variables are similar. 137
8.2 Example of result, here Situation 1 with the Lasso with the modified BIC criterion. Top figure : evolution of the four indicators (recall, precision, Fscore and emptiness) with 95% bandwidth confidence interval in function of c_0. Bottom : the distribution of the precision among all non-empty models for the highest, an intermediate, and the lowest c_0. 143
8.3 Precision in function of emptiness for all tested method for Situation 1. The selectBoost algorithm is compared to both stability selection and the naiveSelectBoost algorithm. 144
8.4 The proportion of correctly identified variables is plotted in function of the confidence index defined in Equation (8.11). The proportion of correctly identified variables is calculated for all variable with a confidence index greater than those mentioned in abscissa. As expected, the greater the confidence index, the higher the proportion of correctly identified variables. 144
8.5 Colors : the green is for the most reliable variables selected by the selectBoost algorithm (confidence index of 0.65 ; orange is for intermediate confidence index (0.55) and red for low confidence (0.45)). Left : evolution of the coefficients in the lasso regression when the sparsity parameter lambda is varying. Right : evolution of the probability of being in the support of the regression when the confidence index is varying. The dotted line represents the threshold of 0.95. The confidence index is calculated as the abscissa at which the probability of being in the support of a variable goes for the first time bellow this threshold. 146

9.1 Nous cherchons des couples de gènes qui ne sont pas exprimés chez les individus sains mais qui forment un couple cible marqueur chez les individus malades. De plus, la cible de ce couple doit être un régulateur important pour le marqueur du couple dans le réseau des individus malades. L'idée est ensuite de faire une expérience d'inhibition du gène cible 1 chez les individus malades, en espérant une réduction significative du gène marqueur 2 chez ces mêmes individus. Si cela se produit, et si aucune modification non prévue dans les expressions des autres gènes est observée, nous serons parvenus à moduler le système malade vers le système sain. 150
9.2 Méthode d'obtention des six listes. 151
9.3 Histogramme des corrélations entre les expressions de gènes obtenues par les méthodes de normalisation dChip et RMA . 152

A.1 Venn Diagram : distribution of the 960 probe sets between the 3 cell groups. A total of 960 probe sets was retained for all the subjects across the three different cell groups. A core of 183 probe sets is shared by the 3 groups. 162
A.2 Each graphic represents genes in a specific categorical time label (1 to 4, from left to right) and their connections, showing how the signal is spreading through the aggressive network. . 163
A.3 DUSP1 is the targeted gene for the knock-down experiment. We show its expression before and after the inhibition experiment. 163
A.4 Principle of the validation experiment. Graphic (a) represents a gene expression before inhibition of a targeted gene. Graphic (b) shows how this gene expression evolves after silencing this targeted gene whereas graphic (c) shows the predicted gene expression. For these two last graphics, for time t_2 to t_4 we assigned a +,- or = label if gene expression after silencing is respectively greater, smaller, or equal to gene expression before silencing. For this gene, graphic (d) shows that we made two good predictions out of three in this example. . . . 163

A.5 Schematic representation of specific constraints related to prediction abilities in model inference. This ability to predict the transcriptional effect of a modulation in the network is crucial in order to predict a gene expression level modification after a knock-down experiment. For instance, given a situation where a gene A regulates the expression of a gene B (with a time lag between activation of gene A and gene B as schematically shown in (a)), which in turn regulates gene C, we want to predict the absence of link between B and C if gene A is knocked-down. Importantly, this predictive capacity requires much more complexity than inference alone. More than inferring a network topology, a predictive method should be able to learn how the biological signal spreads in this network. To go further, the best algorithms for reverse-engineering are not necessarily the best methods for predicting purposes, as explained in (b) with two simple examples. In the first example, a real network is composed of a gene A that activates a gene B, which in turn activates gene C (upper-left quadrant). An inference method could infer a statistical link between A and C, leading to two false negative links (two existing links are not presents) and one false positive link (upper-right quadrant). However, to predict gene C's expression, given the expression of gene A, this inference method will probably give adequate results. In the 2nd example (lower-right quadrant), a better inference method could give six true positive inferred links and only one false negative, omitting the link between A and B. However, in this case, we have a dramatic situation for prediction purpose as gene A cannot activate gene B anymore. 164

A.6 Significance of selected patterns in the clustering step. To evaluate the relevance of our selected patterns used for enrichment, we compared these patterns with various temporal gene clusters obtained with a gold standard unsupervised clustering method. One of the most widely used clustering methods is fuzzy c-means . The preponderant aspect of this algorithm relies on the fuzzy parameter that allows taking into account the inherent noise of transcriptional data (when this parameter increases, more genes are randomly assigned into clusters). For comparison purposes, we focused on the biological data set of patients with more aggressive CLL type and we first select relevant genes with Limma , using a p-value of 0.01. An unsupervised temporal clustering of the 8,113 genes retained with Limma is then performed showing 16 distinct clusters. Importantly, these clusters emphasize the existence of genes with transient expressions (peaks) at t1 (cluster#1, 7, 9), at t2 (within cluster#2, 4), at t3 (cluster#2, 3, 4, 11, 13, 15) and t4 (cluster#5, 6, 8, 10, 12, 14, 16) ; as shown by our method. The fact that through this unsupervised clustering method we reach similar patterns that those produced by our method, confirms the pertinence of our own gene selection process. 165

B.1 Cascade networks are temporal nested networks 168
B.2 Correlation between subjects 172
B.3 Correlation between genes 172
B.4 Correlation structure of the final selection 176
B.5 The F matrices ; for each matrix, the first bar plot corresponds to the coefficient of the diagonal, the second to the first sub-diagonal... 180
B.6 The resulting network with all edges 181
B.7 The resulting network with a cutoff of 0.15 183
B.8 Evolution of scale-freeness of the network in function of the cutoff. The p-value corresponds to the adequacy of the data to a power law distribution. 183
B.9 Neighborhood of gene EGR1 186
B.10 Perturbation modulation at time point 2 of the network consecutively to the knock-out of EGR1. 186
B.11 Evolution of the scale-freeness of the network in function of the cutoff 189
B.12 Evolution of F-score in function of the cutoff 189
B.13 Evolution of F-score in function of the cutoff and the number of subject in the study 190

C.1 Selected genes for inference of the TH1 specific network . . . 197
C.2 Result of the reverse-engineering of the TH1 specific network. 199
C.3 New gene selection with a minimal log fold-change set to 0.5 and very specific temporal patterns. 201
C.4 Biological functions of the selected genes. 202
C.5 Result of the reverse-engineering of the second network. . . . 204
C.6 Predicted effects of the knock-out of IFN-G. 206

D.1 Situation 1 207
D.2 Situation 2 207
D.3 Situation 3 208
D.4 Situation 4 208
D.5 Situation 1 209
D.6 Situation 2 209
D.7 Situation 3 209
D.8 Situation 4 210
D.9 Situation 1. The histograms show the evolution of the distribution of the precision. From left to right : $c_0 = 1, 0.79, 0.54$. 211
D.10 Situation 2. The histograms show the evolution of the distribution of the precision. From left to right : $c_0 = 1, 0.79, 0.69$. 212
D.11 Situation 3. The histograms show the evolution of the distribution of the precision. From left to right : $c_0 = 1, 0.9, 0.81$ . . 213
D.12 Situation 4. The histograms show the evolution of the distribution of the precision. From left to right : $c_0 = 1, 0.87, 0.72$. 214

Préface

Trouver de nouvelles méthodes de traitement du cancer est un enjeu majeur du 21ème siècle. En effet, cette pathologie qui recouvre plusieurs réalités est l'une des premières causes de mortalité en France. Par ailleurs, si certains cancers comme le cancer du sein chez la femme ou le cancer de la prostate chez l'homme ont des taux de survie à cinq ans au-delà de 80%, certains cancers (cancer du pancréas) ont des taux de survie beaucoup plus faibles (moins de 15% de survie à 5 ans) (De Angelis *et al.* 2014).

L'objectif de cette thèse consiste à comprendre les mécanismes cancéreux en se servant d'une approche globale de biologie des systèmes complexes (voir Chapitre 1 pour une définition). Plus précisément, notre objectif est de comprendre les différences entre les programmes géniques sains et tumoraux. De cette compréhension, nous essayerons de prédire le comportement du programme génique en présence de perturbations (Chapitres 4 et 5), ouvrant ainsi la perspective d'effectuer des modulations orientées ...

Avant de présenter le plan de cette thèse, j'aimerais relever l'originalité de ce travail. En effet, de formation purement statisticienne, j'ai cherché à résoudre, par le biais des mathématiques, des problématiques de biologie. Plus précisément, c'est par une collaboration étroite et par des comparaisons systématiques entre résultats d'expériences biologiques et prédictions de mes modèles statistiques que nous sommes parvenus à des résultats significatifs (voir Figure 1). C'est ainsi que nous avons pu, par exemple, définir un nouveau de type de réseau - les réseaux en cascade - et proposer une modélisation statistique adéquate. Le Chapitre 8 propose cependant un outil statistique qui peut être utilisé dans un contexte très général qui dépasse nettement le cadre de la biologie.

Le plan de la thèse est le suivant.

La première partie, intitulée "Introduction", va nous permettre d'exposer les concepts nécessaires dans la suite de la thèse. Plus précisément, cette partie est composée de trois chapitres :

— Le *premier chapitre* expose le cadre général de notre travail de recherche qui est la biologie des systèmes complexes. Nous reviendrons sur les concepts et les définitions des systèmes complexes avant de préciser en quoi la biologie est un champ d'application privilégié des systèmes complexes.
— Le *deuxième chapitre* nous permet d'exposer une méthode statistique

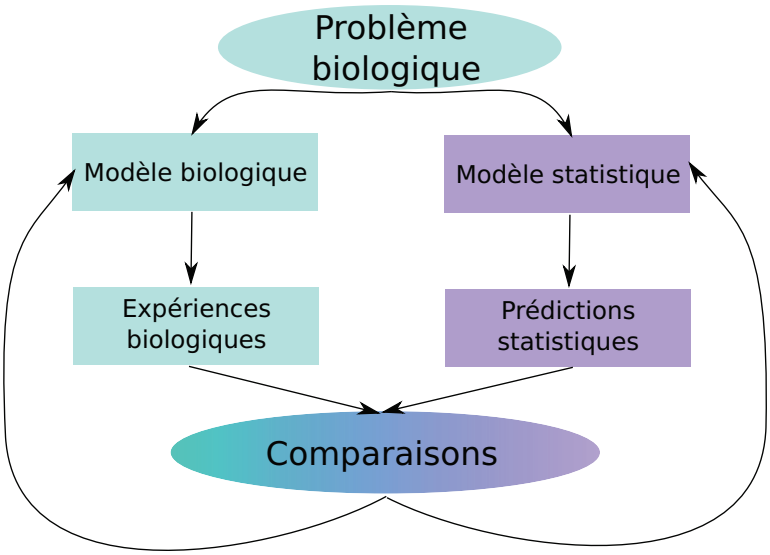


FIGURE 1 – *Notre processus de travail, qui s'inscrit parfaitement dans le cadre de la biologie des systèmes complexes, est le suivant : un problème biologique est posé dans un premier temps ; face à ce dernier, deux travaux s'engagent en parrallèle. D'une part, nous trouvons le travail du biologiste, dont le but est de créer un modèle biologique censé contraindre le problème biologique dans un cadre contrôlable, avant de procéder aux expériences nécessaires. D'autre part, le mathématicien a pour rôle de proposer un modèle statistique adapté au modèle biologique, avant de se servir de ce dernier pour prédire les résultats des interventions des expériences biologiques. Une comparaison doit ensuite être effectuée entre prédictions mathématiques et résultats d'expériences biologiques. Si ces résultats ne sont pas convaincants, il convient pour le biologiste d'affiner son modèle et de procéder à de nouvelles expériences, et au mathématicien d'affiner son modèle statistique et procéder aux nouvelles prédictions. Ce processus doit être répété jusqu'à ce que les comparaisons entre prédictions et résultats d'expérience coïncident.*

dont nous nous servirons tout au long de cette thèse : la régression Lasso.
— Le *troisième chapitre* nous sert à exposer le modèle biologique que nous avons utilisé, ainsi qu'à présenter brièvement le jeu de données sur lequel nous avons majoritairement travaillé dans cette thèse.

La deuxième partie, intitulée "Vers une modulation orientée dans un programme génique" présente nos contributions majeures. Cette partie est composée de trois chapitres augmentés d'annexes :

— Les *quatrième et cinquième chapitres* présentent les travaux publiés dans la revue *PNAS* (Vallat *et al.* 2013). Dans ces travaux, nous prouvons qu'il est possible de prédire les effets d'une modulation dans un programme génique.
— Les *sixième et septième chapitres* exposent les travaux publiés dans la revue *Bioinformatics* (Jung *et al.* 2014). Dans cet article, nous améliorons et généralisons la méthode utilisée dans l'article précédent (Vallat *et al.* 2013). Des améliorations significatives de la méthodologie sont proposées ainsi qu'une implémentation sous la forme d'une librairie (Cascade) pour le logiciel libre de statistique R.
— Le *huitième chapitre* a pour objet des travaux qui sont en cours de soumission . Nous présentons un algorithme dont l'objectif est d'améliorer la précision des méthodes de sélection de variables. L'objectif d'un tel développement est la possibilité de sélectionner avec une grande précision des cibles géniques sur lesquelles intervenir.

La troisième partie, intitulée "Perspectives et conclusions", nous permettra de faire un bilan de nos travaux. Nous l'avons décomposé deux chapitres :

— Le *neuvième chapitre* présente une partie des travaux qui sont en cours à la fin de cette thèse, et qui constituent tout naturellement les prolongements de celle-ci. En particulier, nous présenterons la procédure de choix des cibles pour une intervention orientée dans le programme génique.
— Le *dixième chapitre* proposera un bilan complet des contributions scientifiques apportées par nos travaux.

Par ailleurs, cette thèse a fait l'objet de quatre communications :

— Deuxième colloque international BIO-SI en biostatistiques, Rennes 2011. Communication orale avec pour sujet : *Multistate gene regulatory network inference.*
— Journées de la Société Française de Statistique, Bruxelles 2012. Communication orale avec pour sujet : *Inférence conjointe dans les réseaux de gènes* ; voir Annexe F.
— Deuxièmes rencontres R, Lyon 2013. Poster avec pour titre : *Cascade : un package R pour étudier la dispersion d'un signal dans un réseau de gènes.*
— useR !, Los Angeles 2014. Poster avec pour titre : *Cascade : a R -package to study, predict and simulate the diffusion of a signal*

through a temporal gene network ; voir Annexe E.

D'autre part, durant ma thèse, j'ai apporté ma contribution à divers projets, dont certains sont sur le point d'être soumis à des revues de journal :

— Transcriptomics of circulating human eosinophils unveils activation of a common, cross-disease, immunogenic program (article en cours de soumission).
— Étude de l'impact d'un marqueur génétique dans le cadre de la greffe de mœlle osseuse (article en cours d'écriture).
— Étude de l'impact d'un marqueur génétique dans le cadre de la greffe de rein.
— Dans le cadre d'une participation à la SEME 2014 qui s'est déroulée à Strasbourg : analyse et classification de courbes de charge électriques.

Nous reviendrons dans la conclusions sur les outils et les compétences qui ont été mis en œuvre dans le cadre de ces projets annexes.

Une dernière et quatrième partie comporte les annexes.

Première partie

Introduction

1 Biologie des systèmes complexes

Le titre de cette thèse : "Modélisation de phénomènes biologiques complexes : application à l'étude de la réponse antigénique de lymphocytes B sains et tumoraux" donne à la fois le cadre général dans lequel s'inscrit notre travail et le sujet plus spécifique qui y a été traité. Dans cette introduction, notre but est d'expliquer le cadre général avant de proposer un état de l'art de ce domaine. Le contexte particulier de cette étude sera traité dans le Chapitre 2.

Ce cadre général, nous le définissons comme étant la "biologie des systèmes complexes", terme qui a été introduit au début des années 2000 (Kitano 2000; 2001). Pris séparément, ces trois termes de "biologie", "système" et "complexe", sont facilement définissables. Le dictionnaire Larousse (Collectif 2008) donne à titre d'exemple :

— Biologie : ensemble de toutes les sciences qui étudient les espèces vivantes et les lois de la vie.
— Système : ensemble organisé de principes coordonnés de façon à former un tout scientifique ou un corps de doctrine ou ensemble d'éléments considérés dans leurs relations à l'intérieur d'un tout fonctionnant de manière unitaire ou encore ensemble de procédés, de pratiques organisées, destinés à assurer une fonction définie.
— Complexe : ce qui est complexe, composé de plusieurs parties ou de plusieurs éléments.

Nous pourrions nous attarder plus précisément sur chacun de ces mots, et cela nous conduirait sans aucun doute à parler de complexité au sens de Kolmogorov, à détailler de manière plus précise ce que recouvre le terme système d'un point de vue épistémologique... Ce serait oublier que des mots, mis soigneusement côte à côte, peuvent revêtir une signification à la fois plus large et plus précise que lorsqu'ils sont laissés séparés. Par exemple, il est délicat de comprendre ce qu'est "l'horloge interne" dans le corps humain, en ouvrant le dictionnaire aux mots "horloge" et "interne". C'est pourquoi nous allons commencer cette introduction en tentant de définir ce qu'est un "système complexe" et comment de tels systèmes peuvent être utiles dans le cadre d'études biologiques.

1.1 Systèmes complexes

Avant de définir de manière précise (si tant est que cela soit possible...) ce qu'est un système complexe, nous donnons ici un extrait du livre *La science et l'hypothèse* d'Henri Poincaré (Poincaré 1898) :

> *"Ne pouvons-nous nous contenter de l'expérience toute nue ?*
>
> *Non, cela est impossible ; ce serait méconnaître complètement le véritable caractère de la science. Le savant doit ordonner ; on fait la science avec des faits comme une maison avec des pierres ; mais une accumulation de faits n'est pas plus une science qu'un tas de pierres n'est une maison.*
>
> *Et avant tout le savant doit prévoir.*
>
> *(...)*
>
> *Tel est donc le rôle de la physique mathématique ; elle doit guider la généralisation de façon à augmenter ce que j'appelais tout à l'heure le rendement de la science. Par quels moyens y parvient-elle, et comment peut-elle le faire sans danger, c'est ce qu'il nous reste à examiner."*

Ce que décrit ici Poincaré est la nécessité pour la science d'être gouvernée par des systèmes. Le système est alors vu comme une conceptualisation d'un phénomène permettant de prédire son comportement dans le futur. Mais cela n'est malheureusement pas suffisant pour définir la nature d'un système complexe ; au mieux, la définition d'Henri Poincaré nous permettrait de décrire un système compliqué. Dans la suite nous déterminerons cette différence (système complexe, système compliqué), et expliquerons au lecteur pourquoi un vol d'étourneaux est un système complexe et pourquoi une voiture n'en est pas un.

Intuitivement, nous serions tentés de dire qu'il y a système complexe dès lors que l'ensemble des éléments est plus que la somme de ces derniers. Considérer un système complexe, c'est admettre que dès lors que tous les éléments nécessaires sont rassemblés, ils obéissent ensemble à des lois qui n'étaient pas pré-existantes (c'est-à-dire à des lois en vigueur pour ces éléments pris séparément). Malheureusement, il n'existe pas de définition canonique et rigoureuse de ce qu'est un système complexe, et chaque auteur propose sa définition personnelle. Sans se contredire, chacune de ces définitions insiste sur un point particulier. Nous en donnons quelques-unes ci-dessous :

- Il y a complexité lorsque nous sommes en présence d'une structure assujettie à des variations (Goldenfeld et Kadanoff 1999).
- Un système complexe est un système qui est particulièrement sensible aux conditions initiales ou à de faibles perturbations (Whitesides et Ismagilov 1999).
- Dans un système complexe, le nombre de composantes indépendantes en interaction est important (Whitesides et Ismagilov 1999).

— Dans un système complexe, le système peut évoluer selon différents chemins (Whitesides et Ismagilov 1999).
— Les systèmes complexes sont des systèmes avec de multiples composantes en interaction dont le comportement global ne peut pas être inféré à partir du comportement de chacun des éléments (définition du New England Complex System Institute) .
— Un système est un système complexe dès lors qu'il est composé d'un grand nombre d'éléments en interaction et que la dynamique de ces interactions dirige le comportement du système en lui donnant une apparence d'unité aux yeux d'un observateur. (Beslon et Morange 2008).

Les deux dernières définitions permettent facilement de distinguer un système complexe d'un système compliqué (ce qui, nous le pensons, est la source principale de confusions). En effet, un système compliqué (comme la voiture, par exemple) peut se comprendre à partir de la connaissance de tous les éléments qui le composent. En revanche, un système complexe (le vol des étourneaux par exemple) est intrinsèquement lié aux interactions des éléments qui le composent, et par conséquent, la connaissance simple de chaque élément ne suffit pas à la compréhension de l'ensemble. Certains auteurs parlent également de propriétés émergentes pour décrire les propriétés d'un système qu'il serait impossible de trouver en considérant les éléments séparément (Morin 2013).

Pourquoi donc le vol des étourneaux est-il un système complexe [1] ? D'une part, parce que le groupe composé d'une multitude d'étourneaux a, dans son ensemble, un comportement qui présente une apparence d'unité au regard du spectateur. Ensuite, contrairement à ce qui avait été longtemps supposé, il est impossible de comprendre le comportement du groupe d'étourneaux par une théorie où tous les étourneaux suivraient un chef de file. Au contraire, chaque modification de trajectoire de l'ensemble des étourneaux peut être liée au comportement d'un étourneau. Nous retrouvons donc l'aspect de structure assujettie à des variations en même temps que la sensibilité particulière aux conditions initiales. Ce système complexe, pris ici en exemple, a fait l'objet d'une publication (Cavagna *et al.* 2010).

Ces dix dernières années, les systèmes complexes sont devenus un enjeu de recherche majeure dans le monde, et en France tout particulièrement. Deux instituts majeurs de systèmes complexes y ont vu le jour : l'institut des systèmes complexes de Paris Ile-de-France et l'institut Rhône-Alpin des systèmes complexes. Notons également la présence d'un Réseau National des Systèmes Complexes chargé de mettre en relation les différents acteurs de l'étude des systèmes complexes.

Les systèmes complexes ayant été définis, nous pouvons maintenant nous demander dans quelle mesure ces derniers peuvent s'appliquer à la biologie.

1. `https://www.youtube.com/watch?v=e86-A3DUe9k`

1.2 Biologie des systèmes complexes

La biologie des systèmes complexes vise à une compréhension holistique des systèmes en biologie (Kitano 2002b). L'idée d'une approche globale des systèmes biologiques n'est évidemment pas nouvelle. Par exemple, le principe d'homéostasie selon lequel toutes les variables biologiques d'un être vivant agissent entre elles pour maintenir certains équilibres internes (la régulation de la température peut ici être citée en exemple) a été introduit dès le dix-neuvième siècle (Bernard 1865, Cannon 1932). Il est commun de représenter certains de ces systèmes sous la forme de réseaux (réseaux d'interactions géniques, réseaux d'interactions protéomiques ou réseaux métabolomiques...). Mais cette vision statique et figée du système ne peut pas le décrire et le caractériser entièrement. Nous reprenons ici la vision de Kitano de la description d'un système complexe dans le cadre biologique (Kitano 2002a) :

1. recherche et description du système,
2. compréhension de la dynamique du système,
3. contrôlabilité du système et modification orientée,
4. redéfinition du système.

Nous notons que cette vision s'inscrit pleinement dans le cadre des systèmes complexes que nous nous sommes attachés à définir dans le paragraphe précédent. Le premier niveau de compréhension, la recherche de description du système, revient à décrire les interactions entre les différentes molécules. À ce point, le système est supposé être stable et dans un état spécifique. La dynamique du système permet ensuite de comprendre comment le système sort de son état d'équilibre en fonction des différentes stimulations et perturbations. L'étape suivante consiste à se servir de la compréhension du système et de sa dynamique dans un objectif de contrôle. En effet, une fois la dynamique du système révélée, il est possible de prévoir l'effet d'une modulation du système. Cette contrôlabilité du système permet de changer la façon de concevoir des traitements ; en effet, il n'est plus nécessaire de tester une grande variété de molécules et de sélectionner celle qui a le meilleur effet, mais les prédictions faites à partir de la modélisation du système doivent permettre de déterminer quelle molécule aura l'effet le plus bénéfique. La redéfinition du système est l'étape ultime dans laquelle nous allons chercher à reprogrammer la totalité d'un système pour le faire agir d'une façon différente : par exemple, les bactéries et les levures peuvent être utilisées et reprogrammées afin qu'elles produisent des molécules spécifiques (Hasty *et al.* 2002).

Alors que l'intérêt de considérer les systèmes biologiques comme des systèmes complexes est évident, tant par le nombre d'éléments mis en jeu (par exemple, il y a 10^{14} cellules dans un corps humain) que par leurs interactions, et l'adaptabilité du corps humain aux différentes situations dans lesquelles il est placé, cette vision ne s'est imposée que depuis le début du XXIème siècle. Les causes, nous semble-t-il, peuvent se résumer en deux raisons majeures qui sont :

1. la capacité de la biologie expérimentale à mesurer rapidement et à moindre coût un grand nombre de molécules (voir par exemple l'évolution du coût d'un séquençage d'un génome Figure 1.1). Dans la suite, nous appellerons "techniques de mesure à haut débit" l'ensemble des procédés biologiques permettant de mesurer simultanément un grand nombre de molécules (par exemple, les puces à ADN permettent de mesurer simultanément la production de milliers d'ARNm),
2. la capacité informatique couplée au développement de modèles mathématiques et physiques adaptés pour traiter des jeux de données dont la taille est de plus en plus importante.

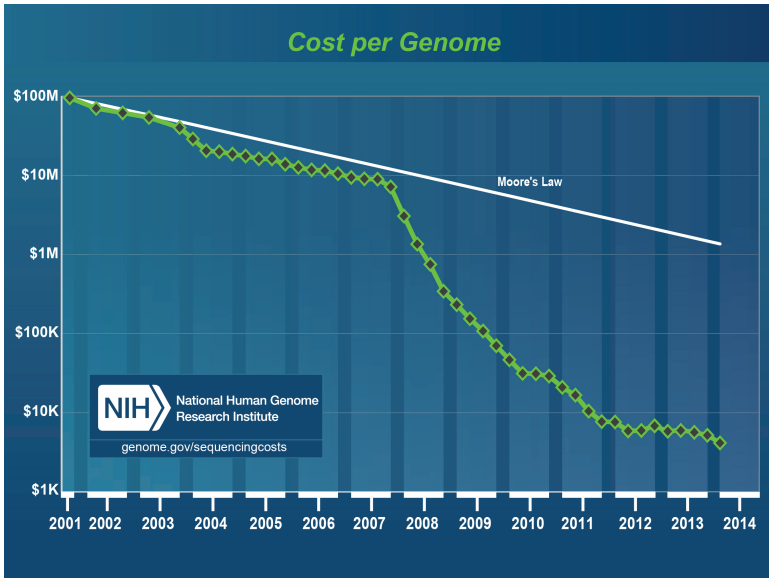


FIGURE 1.1 – *Evolution du prix d'un séquençage complet d'un génome.*

La biologie des systèmes complexes est donc une science entre biologie d'une part et mathématique, physique, bioinformatique, statistique d'autre part... À la première, elle demande des modèles d'étude ainsi que des mesures à haut débit. Aux secondes, elle demande de pouvoir stocker, traiter, analyser, conceptualiser, modéliser les données obtenues. Cette nécessité de dualité est parfaitement illustrée par la Figure 1.2 issue de Ideker *et al.* (2006).

Dans cette thèse, nous nous sommes particulièrement intéressés à l'étude des expressions géniques. Chaque cellule est en elle-même un exemple de système complexe dont nous allons voir qu'il est principalement dirigé par l'expression des gènes.

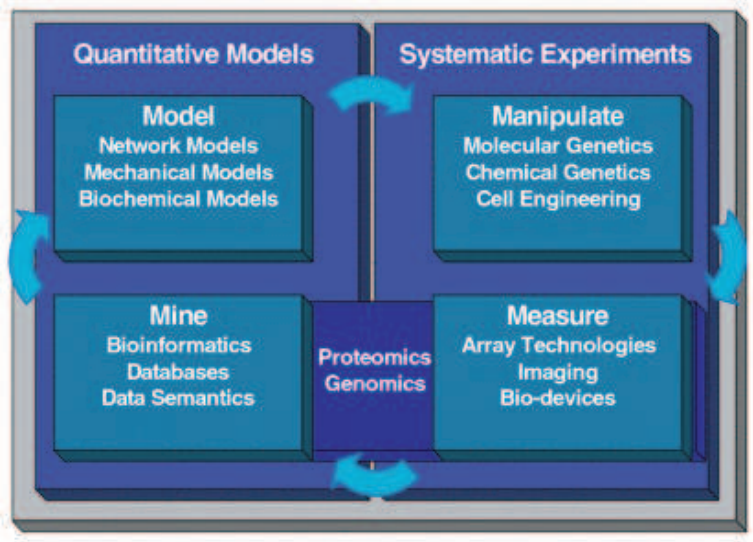


FIGURE 1.2 – *Systèmes complexes en biologie (Ideker* et al. *2006).*

1.3 Réseaux de régulation génique

Toutes les cellules interprètent constamment les messages et les stimuli qu'elles perçoivent pour y apporter une réponse adaptée. Ainsi, durant leur développement, les cellules seront amenées à se différencier, à proliférer ou à synthétiser des hormones spécifiques... Au cours de l'évolution, les cellules ont ainsi développé un grand nombre de processus de régulation dont l'objectif est d'apporter une réponse rapide et spécifique à une situation donnée. Un des mécanismes les plus importants est celui qui contrôle et qui module l'expression du génome, et plus particulièrement les gènes dont la traduction en protéines est à la base de chaque réaction de la cellule.

Pour comprendre le concept des réseaux de régulation génique, il semble nécessaire de revenir aux notions élémentaires, et en particulier au dogme central de la biologie moléculaire tel que défini par Crick (Crick *et al.* 1970). Il peut s'énoncer de la façon suivante : l'acide désoxyribonucléique (ADN) est le support stable et transmissible de l'information génétique qui définit les fonctions biologiques d'un organisme. Il est transcrit en acide ribonucléique (ARN) qui n'a qu'une vie temporaire. L'ARN de type messager (ARNm) est traduit en protéines par les ribosomes. Nous pouvons donc résumer le dogme central de la façon suivante : l'ADN est transcrit en ARN, puis traduit en protéines.

Toutes les parties du génome ne se traduisent donc pas en protéine. Pour donner un ordre d'idée, nous considérons que la partie du génome codant pour des protéines est composée d'environ 25 000 gènes qui couvrent moins de deux pour cent de notre génome. La partie du génome qui ne sera pas traduite par les ribosomes en protéines présente une grande diversité. Nous pouvons citer à titre d'exemple : les ARN ribosomiques, les ARN de transfert, les micro ARN...

Comme nous l'avons annoncé, notre intérêt principal réside dans la compréhension des expressions des gènes. Nous savons que le processus de trans-

cription est gouverné, en majeure partie, par des protéines spécifiques nommées facteur de transcription (FdT) et des ARN non-codants (Sun *et al.* 2012, Walhout *et al.* 2012, Guo *et al.* 2014). Dans le cadre de cette thèse, nous avons été amenés à simplifier cette vue en mettant de côté l'impact potentiel des ARN non codants. Dans ce modèle, l'ADN est transcrit en ARNm qui est traduit en protéines. Les facteurs de transcription peuvent alors agir en retour sur l'activité génique, conduisant à un réseau de régulation génique. Ceci est illustré dans la Figure 1.3.

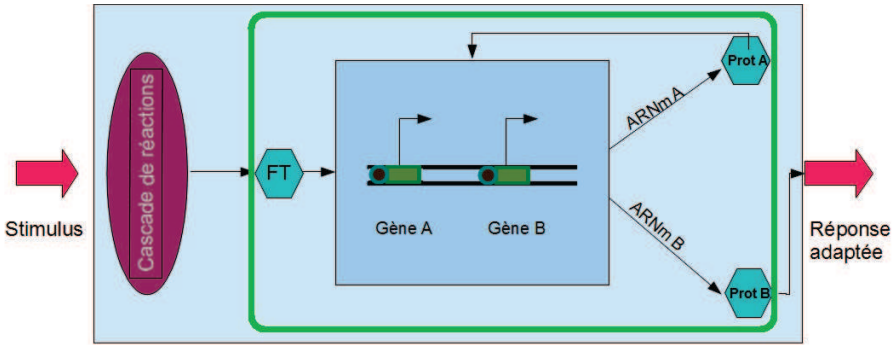


FIGURE 1.3 – *Résumé du principe du réseau de régulation génique.*

Un réseau, tel que décrit ci-dessus d'un point de vue biologique, trouve dans les mathématiques une représentation canonique sous la forme de graphe. Il existe une littérature importante traitant des graphes et de leurs propriétés. Nous n'avons pas l'ambition dans ce manuscrit de couvrir l'ensemble de la théorie des graphes, mais nous souhaitons en donner quelques éléments nécessaires à la compréhension de cette thèse. Dans un soucis d'homogénéité, nous utiliserons le mot "réseau" à la fois dans le contexte mathématique et dans le contexte biologique.

1.4 Généralités sur les réseaux

1.4.1 Définition d'un réseau

Afin de lever une possible ambiguïté, nous distinguerons nettement le programme transcriptionnel et le réseau de régulation de gènes. Ce dernier doit être compris comme étant une modélisation du premier. Définissons tout d'abord ce qu'est un réseau :

Définition 1 (Réseau) *Un réseau* $\boldsymbol{R}$ *est la donnée d'un ensemble discret de nœuds* V *et d'une application* $\gamma : V \times V \to \mathbb{R}$ *qui à chaque couple de nœuds associe un réel. Lorsque* $\gamma(V_1, V_2) \neq 0$*, nous dirons qu'il existe un arc du nœud* V_1 *vers le nœud* V_2*.*

Quelques définitions supplémentaires peuvent caractériser simplement un réseau :

Définition 2 (Réseau non-orienté) *Un réseau* $\boldsymbol{R}$ *est dit non orienté si quel que soit le couple* (V_1, V_2) *de nœuds du réseau* $\boldsymbol{R}$*, nous avons :*

$$\gamma(V_1, V_2) = \gamma(V_2, V_1).$$

Définition 3 (Réseau non-pondéré) *Un réseau $\boldsymbol{R}$ est dit non pondéré si quel que soit le couple (V_1, V_2) de nœuds du réseau $\boldsymbol{R}$, nous avons :*

$$\gamma(V_1, V_2) \in \{0, 1, -1\}.$$

1.4.2 Représentation d'un réseau

Il y a deux représentations "classiques" de réseaux : soit sous forme d'une matrice, appelée matrice d'adjacence, soit sous la forme d'une représentation graphique. Dans la représentation matricielle, nous posons $\boldsymbol{\Omega}$ la matrice d'adjacence du réseau $\boldsymbol{R}$. Dans cette matrice, chaque ligne et chaque colonne représente respectivement le même nœud. Le cœfficient ω_{ij} représente alors le poids affecté à l'arc allant du nœud i sur le nœud j. Autrement dit :

$$\omega_{ij} = \gamma(V_i, V_j).$$

Nous donnons en exemple une matrice d'adjacence pour un réseau avec trois nœuds et quatre arcs :

$$\boldsymbol{\Omega}_1 = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix} \tag{1.1}$$

Les valeurs dans la matrice ont été choisies arbitrairement. Elles appartiennent à $\mathbb{R}$. Cette matrice se lit donc comme suit :

— Le poids de l'arc reliant le nœud 1 à lui-même est égal à 1,
— Le poids de l'arc reliant le nœud 1 au nœud 2 est égal à 2,
— Le poids de l'arc reliant le nœud 2 au nœud 3 est égal à 1,
— Le poids de l'arc reliant le nœud 3 au nœud 1 est égal à -1.
— Le poids des autres arcs est nul.

Une représentation équivalente de ce réseau peut être faite sous la forme graphique. Nous montrons les représentations graphiques du réseau $\boldsymbol{\Omega}_1$ dans la Figure 1.4.

FIGURE 1.4 – *Représentation d'un réseau sans et avec direction (resp. figure de gauche et de droite).*

À partir de là, nous pouvons essayer de rajouter plusieurs types d'information sur le graphique. Par exemple, nous pouvons représenter différemment les poids positifs et les poids négatifs. Ici, nous choisissons de mettre

en rouge les arcs dont le poids est négatif, et en vert les arcs dont le poids est positifs. Nous pourrions aussi épaissir le trait de l'arc en fonction de la valeur absolue du poids de ce dernier (voir Figure 1.5)

FIGURE 1.5 – *Représentation d'un réseau avec des informations supplémentaires (nature du lien à gauche, nature du lien et indicateur de confiance à droite).*

1.4.3 Caractérisation d'un réseau

Pour caractériser un réseau, plusieurs indicateurs peuvent être calculés. Posons d'abord les deux définitions suivantes :

Définition 4 (Degré entrant d'un nœud) *Soit* $\boldsymbol{R}$ *un réseau avec* N *nœuds. Le degré entrant du nœud* V_k*, noté* $d_{ent}(V_k)$ *est défini par :*

$$d_{ent}(V_k) = \sum_{n=1}^{N} \mathbf{1}_{\gamma(V_n,V_k)\neq 0},$$

où $\mathbf{1}$ *est la fonction indicatrice.*

Définition 5 (Degré sortant d'un nœud) *Soit* $\boldsymbol{R}$ *un réseau avec* N *nœuds. Le degré sortant du nœud* V_k*, noté* $d_{sort}(V_k)$ *est défini par :*

$$d_{sort}(V_k) = \sum_{n=1}^{N} \mathbf{1}_{\gamma(V_k,V_n)\neq 0},$$

où $\mathbf{1}$ *est la fonction indicatrice.*

Dans le cas d'un réseau non-orienté nous avons la propriété suivante :

Propriété 6 *Dans le cas d'un réseau non-orienté, les degrés entrant et sortant d'un nœud sont égaux.*

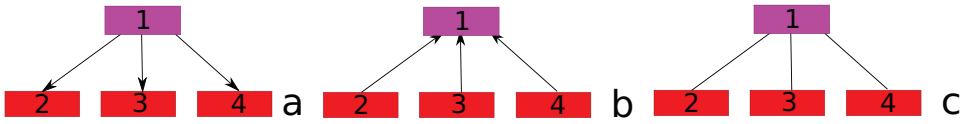


FIGURE 1.6 – *a : le degré sortant du nœud 1 est de 3. b : le degré entrant du nœud 1 est de 3. c : les degrés sortant et entrant du nœud 1 sont de 3.*

Comme nous le verrons dans un paragraphe suivant, la forme de la distribution des degrés entrant et sortant des nœuds est très utile pour caractériser un réseau.

Avant de poursuivre, nous devons définir la notion de chemin dans un réseau :

Définition 7 (Chemin) *Un chemin P entre deux nœuds V_i et V_j est un ensemble ordonné de nœuds $P = (V_i, V_{k_1}, ..., V_{k_K}, V_j)$ tels que $\gamma(V_i, V_{k_1}) \neq 0$, $\gamma(V_{k_1}, V_{k_2}) \neq 0$, ... , $\gamma(V_{k_K-1}, V_{k_K}) \neq 0$, $\gamma(V_{k_K}, V_j) \neq 0$.*

La longueur d'un chemin est définie par $K+1$ lorsque le réseau n'est pas pondéré et par :

$$|\gamma(V_i, V_{k_1})| + |\gamma(V_{k_1}, V_{k_2})| + ... + |\gamma(V_{k_{K-1}}, V_{k_K})| + |\gamma(V_K, V_{k_j})|$$

lorsque que le réseau est pondéré.

Un plus court chemin entre deux nœuds est un des chemins dont la longueur est minimale.

La notion de plus court chemin définie ci-dessus permet d'introduire une distance naturelle entre deux nœuds :

Définition 8 (Distance entre nœuds) *Soit $P = (V_i, V_{k_1}, ..., V_{k_K}, V_j)$ le plus court chemin entre les nœuds V_i et V_j. La distance entre ces deux nœuds, $d(V_i, V_j)$ est alors définie par la longueur du plus court chemin tel que défini ci-dessus.*

Nous pouvons alors définir la notion de diamètre d'un réseau :

Définition 9 (Diamètre d'un réseau) *Le diamètre d'un réseau est égal à la longueur du plus long des plus petits chemins.*

De la même manière il est possible de définir la longueur moyenne d'un chemin :

Définition 10 (Longueur moyenne d'un chemin) *La longueur moyenne d'un chemin d'un réseau est définie par la moyenne des plus petits chemins.*

Les notions de diamètre du réseau et de longueur moyenne d'un chemin sont particulièrement importantes dans les réseaux biologiques, parce qu'un chemin court implique un temps de réponse plus court.

Il existe un autre indicateur qui peut avoir un intérêt pour caractériser les réseaux : il s'agit du cœfficient de clustering. Le cœfficient de clustering est avant tout une mesure locale d'un réseau. Il est définit de la manière suivante :

Définition 11 (Cœfficient de clustering) *Le cœfficient de clustering d'un nœud V_i dans un réseau est défini comme suit (Watts et Strogatz 1998) :*

$$\frac{2e_i}{k_i(k_i-1)},$$

avec k_i le nombre de nœuds qui ont un lien avec V_i et e_i le nombre de liens entre ces nœuds.

Le cœfficient de clustering du réseau entier est égal à la moyenne des cœfficients de clustering pour chaque nœud du réseau.

1.4.4 Caractérisation d'un nœud dans le réseau

Après avoir donné quelques définitions pouvant caractériser le réseau dans son ensemble, nous allons nous attacher à trouver quelques indicateurs pouvant définir un nœud du réseau en particulier. Comme nous venons de le voir, le cœfficient de clustering peut être un de ces critères, permettant de faire la distinction entre les nœuds dont les voisins sont connectés (et l'ensemble formant ainsi un module) et les nœuds dont les voisins sont isolés les uns des autres.

Les indicateurs que nous allons présenter maintenant servent, pour la plupart, à déterminer les nœuds "importants" du réseau. L'importance d'un nœud dans le réseau peut être définie de plusieurs manières. La manière la plus intuitive pour définir un nœud important dans un réseau est de regarder ses degrés entrant et sortant. Un nœud sera alors dit "hub" lorsque ces nombres seront grands (bien qu'il n'existe pas de définition précise d'un nombre minimal pour définir un "hub").

Nous introduisons maintenant deux notions de centralité (Bonacich 1987) la centralité de proximité et la centralité d'intermédiarité. La centralité de proximité permet de savoir si, en moyenne, un nœud est proche ou éloigné des autres nœuds. Un nœud dont la centralité de proximité est grande sera éloigné des autres nœuds et il sera à la fois peu influent et peu influençable par les autres nœuds. A l'opposé, un nœud dont la centralité de proximité est grande est un nœud influent et par lequel la transmission d'une information sera rapide. La centralité d'intermédiarité pour un nœud donné se définit par la proportion parmi tous les autres couples de nœuds, de plus courts chemins qui passent par lui.

Maintenant que nous avons défini ce qu'était un réseau et quelles étaient ses premières caractéristiques, nous allons voir qu'il existe plusieurs topologies classiques pour les réseaux.

1.4.5 Quelques topologies classiques de réseaux

Dans cette partie, nous supposerons que les réseaux dont il est fait référence sont non-orientés. En effet, une fois compris les mécanismes permettant de définir et de classer les réseaux aléatoires non-orientés, il est aisé de définir d'élargir la définition aux réseaux orientés. Notons que dans ce cadre, la notion d'arête se substitue à la notion d'arc.

Maintenant que nous avons défini plusieurs indicateurs permettant de caractériser les réseaux, tant d'un point de vue local que d'un point de vue global, nous allons répertorier les topologies classiques de réseaux aléatoires. Nous opposons ici la notion de réseaux aléatoires à la notion de réseaux structurés et réguliers. Ce qui distingue ces deux types de réseaux est la manière dont ils ont été conçus, c'est-à-dire la manière dont les arêtes du réseau ont été ajoutées. Les premiers sont supposés être le résultat d'un

processus aléatoire ; par exemple, le réseau dans lequel chaque nœud correspond à une personne et chaque arête correspond à un lien d'amitié entre ces personnes. Les seconds sont une conception humaine ; par exemple, un réseau où il existe une arête entre chaque nœud.

Les réseaux aléatoires de Erdös-Rényi

Les réseaux aléatoires proposés par Erdös et Rényi sont les plus simples et les plus intuitifs à concevoir (Erdos et Renyi 1959). Les auteurs proposent deux façons de construire des réseaux aléatoires :

— Il est demandé de choisir une probabilité $0 < p < 1$. Chaque arête potentielle du réseau est alors présente avec une probabilité p.
— Il est ici demandé de choisir le nombre m d'arêtes présentes dans le réseau. Un tirage au sort sans remise de m arêtes parmi toutes celles possibles est alors effectué et détermine les arêtes qui seront présentes dans le réseau.

Nous présentons en Figure 1.7 un exemple d'un tel graphique. Les auteurs ont prouvé plusieurs propriétés de réseaux ainsi construits. Par exemple, lorsque le nombre de nœuds devient grand, le réseau devient presque sûrement connexe. Ce qui va nous intéresser par la suite, ce sont les propriétés suivantes :

Propriété 12 *Soit G un réseau aléatoire de N nœuds créé selon le principe de Erdös et Rényi, avec une probabilité p de présence pour chacune des arêtes. Posons :*

$$\lambda = (N-1)p.$$

Soit D_i le nombre d'arêtes reliées au nœud i. Nous avons alors :

$$\mathbb{P}(D_i = k) = \frac{e^{-\lambda}\lambda^k}{k!}.$$

La distribution du nombre d'arêtes par nœud suit donc une loi de Poisson de paramètre λ. D'autre part, le cœfficient de clustering moyen pour l'ensemble des nœuds d'un tel réseau est p.

Nous nous intéresserons également à la longueur moyenne du plus court chemin. Dans un réseau à 1000 nœuds, nous avons fait varier la probabilité p de présence d'une arête dans le réseau. Le résultat est présenté en Figure 1.8.

Propriété des petits mondes

Lorsque nous parlons de propriété des petits mondes, nous nous référons toujours à l'étude - quoique critiquable - de Milgram. Ce dernier envoya 60 lettres à des recrues de la ville d'Omaha dans le Nebraska. Il leur demanda de faire suivre cette lettre à un agent de change, vivant à une adresse fournie, dans la ville de Sharon dans le Massachusetts. Les participants pouvaient seulement passer les lettres, de main à main, à des connaissances personnelles qu'ils pensaient capables d'atteindre l'objectif, directement ou via les

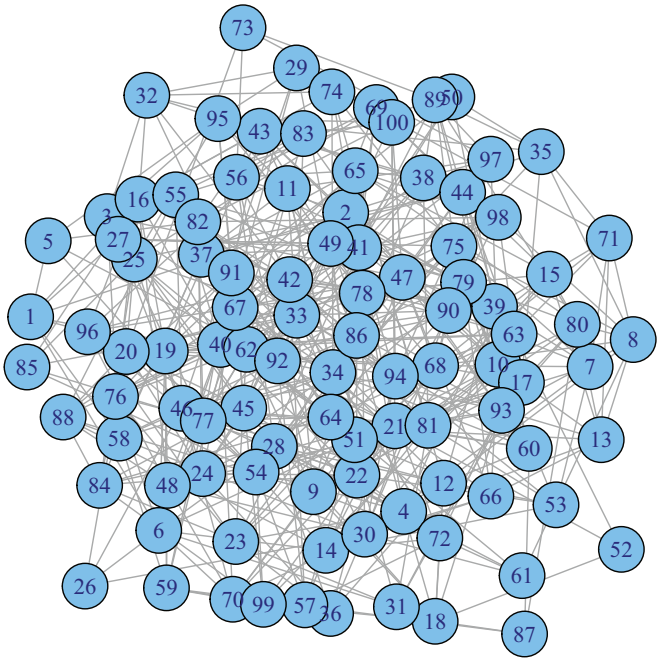


FIGURE 1.7 – *Réseau aléatoire contenant 100 nœuds et 500 arêtes engendré selon le principe d'Erdös et Rényi.*

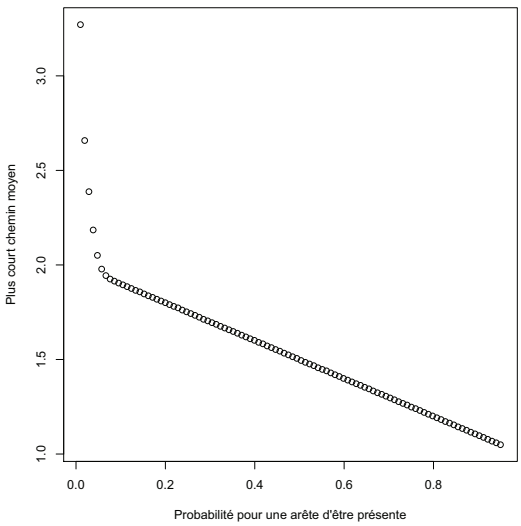


FIGURE 1.8 – *Dans un réseau aléatoire à 1000 nœuds, évolution de la longueur moyenne du plus court chemin en fonction de la probabilité p de présence d'une arête dans le réseau.*

amis des amis. Bien que cinquante personnes se soient prêtées à l'expérience, seulement trois lettres arrivèrent à destination. Le célèbre article de 1967 de

Milgram décrit le fait qu'une lettre ne mit que quatre jours pour atteindre sa destination.

La propriété des petits mondes stipule que dans la plupart des réseaux réels (nous en verrons quelques-uns en détails) le plus court chemin moyen entre deux nœuds est faible, c'est-à-dire proportionnel au logarithme du nombre total de nœuds (Watts et Strogatz 1998). Cette propriété est vérifiée dans le cadre de réseaux aléatoires.

Les réseaux à structure régulière ne jouissent pas de cette propriété. Cependant, du fait de leur construction, ils ont souvent un indice de clustering moyen élevé (moyenne de l'indice de clustering de l'ensemble des nœuds du réseau), ce qui peut être une propriété souhaitable. Pour parvenir à donner la propriété de petits mondes à des structures régulières, Watts et Strogatz (1998) proposèrent de suivre le procédé suivant :

1. Créer un réseau régulier, par exemple en disposant les nœuds en cercle, et en reliant chaque nœud aux deux nœuds précédents.
2. Déterminer une probabilité p.
3. Changer aléatoirement chaque lien avec une probabilité p.

Il peut être montré qu'une telle procédure permet à la fois d'obtenir la propriété des petits mondes et de garder une partie de la structure initiale, conduisant généralement à un cœfficient de clustering élevé. Nous montrons dans la Figure 1.9 la construction d'un tel réseau.

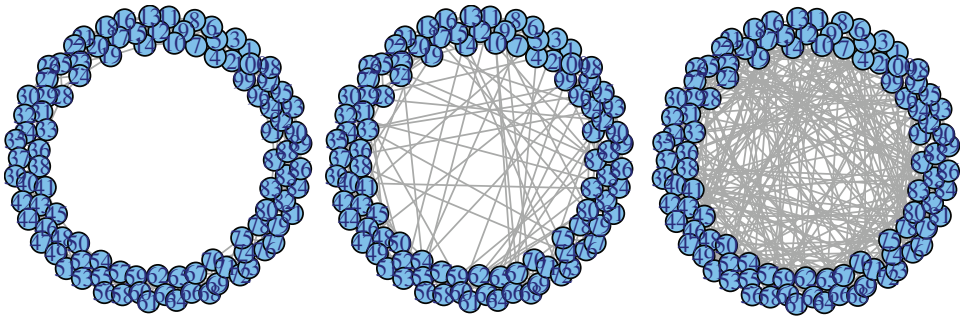


FIGURE 1.9 – *D'une structure régulière à une structure aléatoire. À gauche : structure parfaitement régulière de 100 nœuds. Au milieu : chaque arête est modifiée aléatoirement avec une probabilité de* $p = 0.1$. *À droite : toutes les arêtes sont modifiées aléatoirement,* $p = 1$.

Ce qui est particulièrement intéressant avec la construction de Watts et Strogatz (1998), c'est qu'il suffit de modifier un faible pourcentage de liens pour faire baisser de manière quasiment optimale (*i.e.* : l'optimalité étant obtenue pour un réseau parfaitement aléatoire) la longueur du plus court chemin moyen. Nous avons simulé un exemple dans lequel nous sommes partis d'un réseau régulier à 1000 nœuds où chaque nœud est initialement connecté aux 10 nœuds précédents (structure comparable à la Figure 1.9 à gauche). Nous avons alors commencé à suivre le procédé de Watts et Strogatz. Nous remarquons qu'avec 5% de modification d'arêtes dans le réseau, nous gardons un fort cœfficient de clustering et une distance moyenne des plus courts chemins quasiment optimale (Figure 1.10).

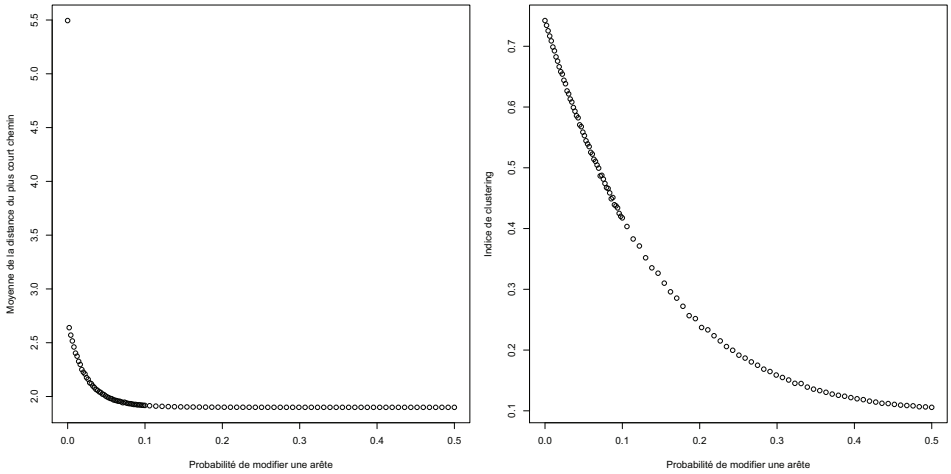


FIGURE 1.10 – *D'une structure régulière à une structure aléatoire, analyse de la moyenne des plus courts chemins moyens et du cœfficient de clustering.*

Réseaux invariants d'échelle

Parmi les réseaux présentant la propriété des petits mondes, le modèle des réseaux invariants d'échelle est un exemple d'un intérêt particulier. Ils sont définis comme suit :

Définition 13 (Réseau invariant d'échelle) *Un réseau G est dit invariant d'échelle (scale-free network, en anglais) si la distribution du nombre de liens par nœud suit une loi de type puissance. Plus précisément, soit D_i le nombre d'arêtes reliées au nœud V_i (voir Figure 1.11 pour un exemple) :*

$$\mathbb{P}(D_i = k) \propto k^{-\gamma}.$$

Ces réseaux ont été découverts pour la première fois par Price (1976) qui étudiait le nombre de citations des publications scientifiques. Sans les nommer réseaux invariants d'échelle, il découvrit que, contrairement à ce qui était connu, la distribution du nombre de citations avait une queue lourde. Il expliqua cette propriété par un principe d'avantage cumulatif : plus un papier est cité, et plus il a de chance d'être cité encore (Price 1976) !

En 1999, Barabási et Albert (1999) ont introduit formellement les réseaux invariants d'échelle, en partant de l'étude du réseau Internet. Depuis, une multitude d'études a prouvé que l'essentiel des réseaux réels (dont font évidemment partie les réseaux biologiques) sont de type invariant d'échelle. Nous donnons quelques exemples dans le Tableau 1.1.

1.5 Inférence de réseaux de régulation génique

Comme nous venons de le voir, les systèmes biologiques sont étonnamment complexes. Alors que la biologie moléculaire a permis de révéler le fonctionnement et les interactions d'une multitude de molécules, la biologie des systèmes complexes apporte la promesse d'une vision holistique de l'ensemble de ces éléments. La modélisation de programmes géniques sous la forme de réseaux de régulation de gènes s'inscrit parfaitement dans cette

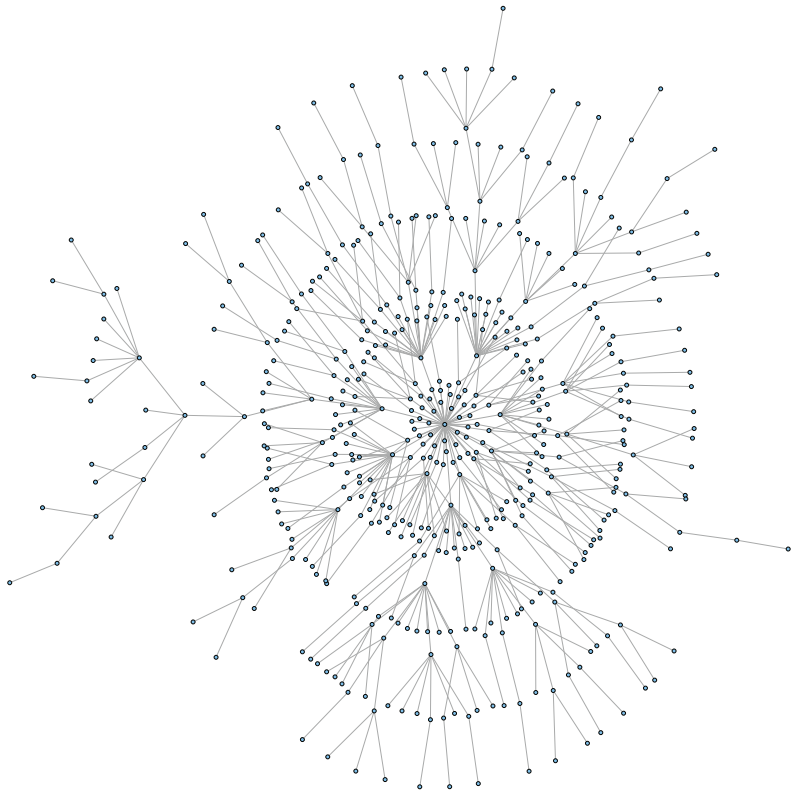


FIGURE 1.11 – *Exemple de réseau invariant d'échelle*

Réseau	Référence	Nœuds	Clustering	Distance moyenne	γ
Internet	Vazquez *et al.* (2002)	228298	0,03	9,51	2,1
Acteurs de film	Barabási et Albert (1999)	225226	0,79	3,65	2,3
Coll. math.	Newman (2001)	70975	0,59	9,5	2,5
Réseau métabolique	Jeong *et al.* (2000)	778	N.C.	3,2	2,2

TABLE 1.1 – *Quelques caractéristiques de réseaux réels : le réseau de routage internet, les participations communes d'acteurs dans les films, les collaborations mathématiques dans les articles publiés et un réseau métabolique. Nous donnons le nombre de nœuds, le cœfficient de clustering, la distance moyenne du plus court chemin, et le cœfficient pour les distributions de type puissance.*

approche. Dans cette thèse, nous utiliserons le principe d'ingénierie inverse pour reconstruire les réseaux de gènes (De Jong 2002). L'ingénierie inverse se propose de modéliser un phénomène à partir d'observations issues de ce

dernier. En effet, les données de départ dont nous disposons sont les quantités d'ARNm présentes dans la cellule pour l'ensemble des gènes à des moments différents. C'est à partir de ces données d'expression génique que nous chercherons à inférer le réseau de régulation des gènes.

1.5.1 Algorithmes et méthodes

Peu d'hypothèses sont faites dans l'inférence de réseaux de gènes. Cependant, l'hypothèse qui nous semble la plus essentielle est l'hypothèse de parcimonie. Plus précisément, un gène est supposé être contrôlé par un nombre limité de régulateurs (Leclerc 2008). Ensuite, les réseaux de régulation génique sont supposés être de type invariant d'échelle. D'autres hypothèses peuvent être nécessaires en fonction du modèle considéré.

Aperçu général

Plusieurs revues ont essayé de faire un état de l'art sur les méthodes d'inférence de réseaux de gènes existantes ; chacune classe les méthodes selon différents critères et porte une attention particulière à certains points précis.

La première revue d'importance sur les méthodes d'inférence de réseaux de gènes a été (De Jong 2002). Plus récemment, Chao *et al.* (2009) ou Hecker *et al.* (2009) donnent un aperçu de l'ensemble des méthodes d'inférence de réseaux de gènes basées sur des données temporelles. Dans le premier, une distinction importante est faite entre les méthodes permettant de décrire la dynamique d'un système temporel, et celles ne le pouvant pas. Décrire la dynamique du système est absolument nécessaire lorsque nous voulons pouvoir faire de la prédiction. Dans un article récent, He *et al.* (2009) mettent en lumière les hypothèses mathématiques et biologiques sous-jacentes à chaque modèle. Les principales méthodes pour analyser les expressions de gènes, tant pour détecter les gènes différemment exprimés que la classification des expressions et l'inférence des réseaux géniques, ont été listées par Bar-Joseph *et al.* (2012).

La liaison statistique entre deux gènes peut être considérée inconditionnellement, conditionnellement à un ensemble de gènes ou conditionnellement à l'ensemble des autres gènes. Cette distinction intéressante est décrite par Markowetz et Spang (2007). Cela est particulièrement intéressant puisque la distinction entre liens directs et indirects est particulièrement difficile dans les réseaux de gènes.

Enfin, certaines revues mettent l'accent sur un type de méthode particulier ; Friedman (2004), par exemple, traite le cas des réseaux bayésiens et chaînes de Markov. Il insiste par ailleurs sur le fait que mesurer les ARNm est potentiellement une source de biais, étant donné qu'il peut y avoir des modifications post-transciptionnelles.

Pour décrire les méthodes existantes, nous nous baserons sur les catégories énoncées par Bansal *et al.* (2007), c'est-à-dire que nous distinguerons :

1. Les méthodes dites “statistiques” ou d’interaction, qui cherchent simplement à mesurer la proximité entre les expressions de gènes,
2. Les méthodes basées sur des équations,
3. Les méthodes dites d’optimisation.

Méthodes d’interaction

Dans les méthodes d’interaction, il faut choisir une mesure de proximité, et un seuil à partir duquel les expressions de deux gènes seront supposées suffisamment proches pour pouvoir supposer qu’il existe un lien entre eux. Notez que nous employons ici le mot “lien” pour décrire un arc ou une arête selon que le réseau inféré est dirigé ou non. Parmi les mesures de proximité utilisées nous pouvons citer le cœfficient de corrélation de Pearson (Schafer et Strimmer 2005), le cœfficient d’information mutuelle (Margolin *et al.* 2006a) ou encore la distance euclidienne (Ruan 2010). Ces méthodes, généralement simples et peu gourmandes en temps de calcul, ne peuvent néanmoins pas décrire la dynamique d’un système temporel.

La méthode présentée par Margolin *et al.* (2006a), nommée ARACNe, est particulièrement intéressante, puisqu’elle a été une des premières à pouvoir reconstituer un très large réseau de gènes. Nous trouvons par ailleurs une version adaptée pour les séries temporelles dans Zoppoli *et al.* (2010), où les expressions temporelles des gènes sont synchronisées en fonction de leur premier pic d’activité. Une méthode équivalente, utilisant également l’information mutuelle a été proposée (Yalamanchili *et al.* 2014). Sans entrer dans les détails, ARACNe se base donc sur le cœfficient d’information mutuelle, qui permet de détecter des similitudes, même non-linéaires, entre les expressions de gènes. La force de cette méthode est de proposer une inégalité spécifique à l’information mutuelle pour réduire le nombre de faux positifs. Cette inégalité, basée sur une approximation, a été améliorée pour donner la méthode hARACNe (Jang *et al.* 2013). De manière plus générale, Villaverde *et al.* (2013) ont écrit une revue sur l’utilisation de l’information mutuelle dans l’inférence des réseaux de gènes.

Une classe de méthode importante dans les méthodes d’interaction sont les modèles graphiques gaussiens (Graphical Gaussian Models, ou GGMs, en anglais). Ils ont été proposés pour la première fois dans le cadre des réseaux de gènes par Kishino *et al.* (2000). Ces modèles ont ensuite été largement repris, en particulier par Schafer et Strimmer (2005) et Chiquet (2011). Nous détaillons ici la méthode proposée par Schafer et Strimmer (2005). Supposons que nous disposons d’une matrice des données $\boldsymbol{X}$ avec N lignes (les répétitions, les patients...) et G colonnes (les gènes). Il est supposé par ailleurs que $\boldsymbol{X}$ provient de la réalisation d’une loi normale multivariée $\mathcal{N}_G(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ avec $\boldsymbol{\mu} = (\mu_1, ..., \mu_G)'$ le vecteur des moyennes, et $\boldsymbol{\Sigma} = (\sigma_{ij})$ la matrice définie positive de variance covariance, pour $1 \leqslant i, j \leqslant G$. Grâce à la formule $\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$, nous pouvons décomposer la matrice $\boldsymbol{\Sigma}$ entre les composantes de variance σ_i et les cœfficients de corrélation de Pearson p_{ij} issus de la matrice de corrélation P. Dans les GGMs, nous nous intéresserons à la corrélation partielle, qui dans le cas de normalité, peut s’obtenir en inversant la matrice P. Dans notre cas, nous avons $N << G$ et par conséquent,

il n'existe pas de telle inverse. Il faut alors utiliser des pseudo-inverses, ou inverses généralisées, comme proposé par Moore et Penrose (Penrose 1955). L'utilisation de cet estimateur est biaisée, et c'est pourquoi Schafer et Strimmer (2005) proposent l'utilisation de méthodes bootstrap pour réduire le biais.

Enfin, certaines méthodes utilisent les méthodes d'interaction comme un *a priori* qu'il s'agit ensuite d'affiner (Ruan 2010).

Méthodes à équations

Dans le cas où il est possible de supposer les interactions linéaires et additives, il semble naturel de poser le modèle suivant :

$$\frac{\boldsymbol{x}(t+1)-\boldsymbol{x}(t)}{\Delta} = \boldsymbol{\Omega x}(t) + \boldsymbol{A b}(t) + \boldsymbol{\epsilon(t)} \tag{1.2}$$

où $\boldsymbol{x}(t)$ est le vecteur d'expression des gènes au temps t, Δ est l'intervalle de temps entre t et $t+1$, $\boldsymbol{\Omega}$ est la matrice d'interaction des gènes, $\boldsymbol{b}(t)$ sont des potentielles perturbations appliquées au système, $\boldsymbol{A}$ une matrice à estimer, et $\boldsymbol{\epsilon(t)}$ le vecteur des erreurs au temps t.

Dans le cas de données statiques, il suffit de supposer $\boldsymbol{x}(t+1) = \boldsymbol{x}(t)$ ou encore poser :

$$\frac{\boldsymbol{x}(t+1)-\boldsymbol{x}(t)}{\Delta} = 0$$

Ce problème se résumerait à un problème de régression classique ; mais dans le cadre des réseaux de gènes, comme nous l'avons déjà vu, le nombre d'observation N est largement plus petit que le nombre de variables G (les gènes). L'approche la plus directe semble être de chercher à utiliser la décomposition en valeurs singulières (Eckart et Young 1936, Yeung *et al.* 2002)(SVD decomposition, en anglais). Une approche similaire peut être trouvée dans l'algorithme TSNI (Bansal *et al.* 2006). Le désavantage d'une telle approche est qu'elle ne permet pas d'obtenir une solution unique. Pour traiter ce problème Yeung *et al.* (2002) proposent une régression robuste $\mathcal{L}_1$ pour choisir la solution optimale parmi les solutions parcimonieuses. Une approche assez similaire est trouvée dans Opgen-Rhein et Strimmer (2007). Les auteurs portent ici une attention particulière à l'estimation de la matrice de variance-covariance, en appliquant une contraction de Stein (James et Stein 1961) (la méthode générale a été proposée par Schafer et Strimmer (2005)). Une réduction de la dimensionnalité par une analyse en composantes principales est utilisée dans la méthode TSNI (Bansal *et al.* 2006).

Dans l'algorithme NIR (Gardner *et al.* 2003) pour chaque gène, le nombre de gènes régulateurs est supposé suffisamment limité pour pouvoir faire une régression classique. Il suffit alors de tester tous les sous-ensembles possibles et choisir celui qui minimise l'erreur quadratique. Il est évident que le désavantage majeur de cette méthode est qu'elle n'est pas applicable à de grands jeux de données (dans ce cas, le nombre de sous-ensembles à tester est alors trop grand). Une version parallélisée a cependant été proposée (Gregoretti

et al. 2010).

Une version bayésienne de régression est proposée par Rogers et Girolami (2005). Cette approche est intéressante du fait que l'estimation du réseau est parcimonieuse par nature, si l'on choisit des densités *a priori* adaptées. Une approche différente, dans laquelle l'ordre de la différence est discuté (*i.e.* le $\delta \in \mathbb{N}$ dans $\boldsymbol{x}(t+\delta) - \boldsymbol{x}(t)$) a été proposée (Bickel *et al.* 2009).

D'un point de vue purement théorique, il s'agit d'un problème de sélection de variables. Une approche intéressante consiste à intégrer la sélection de variables sous forme de contrainte dans la régression. Pour ce faire, la pénalité Lasso (Tibshirani 1996) est sans doute la plus largement utilisée (Christley *et al.* 2009). L'idée est de rajouter une pénalité de type $\mathcal{L}_1$ à la régression. Cette méthode a été d'autant plus populaire lors de l'introduction de l'algorithme LARS (Efron *et al.* 2004) qui permet de trouver l'ensemble du chemin des solutions (ou encore le chemin de régularisation) avec un temps de calcul équivalent à une régression linéaire classique. Mais l'estimateur Lasso, comme souligné par Leeb et Pötscher (2008) par exemple, a plusieurs désavantages. Le plus important, dans le cadre de l'inférence de réseaux de gènes, c'est que parmi un ensemble de prédicteurs potentiels fortement corrélés, l'estimateur Lasso aura tendance à en choisir un seul, et devient, de ce fait, très sensible au bruit. C'est pourquoi plusieurs estimateurs ont été proposés, comme l'estimateur Elastic Net, qui est un mélange de régression Ridge (pénalité $\mathcal{L}_2$) et de régression Lasso (Zou et Hastie 2005). Plus récemment Zhang (2010) propose un estimateur basé sur une pénalité concave réduisant le biais de l'estimateur Lasso au maximum. Une pénalité $\mathcal{L}_2$ a été ajouté dans ce dernier modèle par Huang *et al.* (2011). Des approches différentes et complémentaires sont également possibles : Belloni et Chernozhukov (2011) propose une régression parcimonieuse dans laquelle nous nous intéressons à la médiane de la variable endogène, et Lê Cao *et al.* (2008) proposent une version PLS (Partial Least Square).

Parmi les estimateurs ci-dessus, rares sont ceux à avoir été utilisés en pratique. L'estimateur Lasso a été utilisé par Bonneau *et al.* (2006) ou Christley *et al.* (2009). Ce dernier propose de rajouter d'autres contraintes $\mathcal{L}_1$ afin de prendre en compte l'information *a priori* disponible. L'estimateur Elastic Net a été utilisé par Gustafsson et Hörnquist (2010). Les auteurs tiennent également compte de l'information *a priori* disponible en pondérant les régresseurs.

Afin de lever l'hypothèse de linéarité, deux classes spécifiques de modèles non-linéaires ont été introduites. Il y a d'abord les S-systèmes (Akutsu *et al.* 2000) qui s'écrivent sous la forme :

$$\frac{\boldsymbol{x}(t+1) - \boldsymbol{x}(t)}{\Delta} = \alpha \prod_i^N X_i^{g_i} - \beta \prod_i^N B_i^{a_i}$$

avec $\{\alpha, \beta, g_1, ..., g_N, a_1, ..., a_N\}$ un ensemble de paramètres à estimer. Comme le modèle n'est pas linéaire, nous ne pouvons évidemment pas appliquer les techniques de régression classiques. Pour ce faire, Akutsu *et al.* (2000) proposent un algorithme génétique. La deuxième classe de modèles

non-linéaires est représentée par les réseaux de neurones. L'approche proposée par Xu *et al.* (2007) pour estimer les paramètres d'un tel modèle est celle de l'optimisation par essaims particulaires. Nous simulons des solutions potentielles, et nous les faisons évoluer indépendamment, tout en échangeant à chaque étape l'information des solutions les plus proches. D'autres approches, avec des hypothèses de structure moins fortes, peuvent être trouvées : Kim *et al.* (2004) utilisent une régression non paramétrique, Bonneau *et al.* (2006) dans leur algorithme Inferlator et Gustafsson *et al.* (2009) proposent d'appliquer des transformations non-linéaires des prédicteurs.

Pour terminer, nous citerons Ahmed et Xing (2009) qui proposent d'inférer un réseau dont les liens évoluent avec le temps grâce à une régression logistique avec une pénalité Lasso ; une contrainte $\mathcal{L}_1$ supplémentaire y est proposée pour que la variation des réseaux entre deux temps ne soit pas trop forte.

Méthodes d'optimisation

Il existe deux grandes familles de modèles d'optimisation :

1. Les Réseaux Booléens (RBo), et les Réseaux Multi-états (RM)
2. Les Réseaux Bayésien (RB) et les les Réseaux Bayésiens Dynamiques (RBD)

Nous allons traiter ces deux grandes familles séparément.

Réseaux Booléens et Réseaux Multi-états Dans les deux cas, il faut discrétiser les variables. Dans un RBo, on supposera qu'un gène a un comportement binaire, soit actif, soit inactif. Les RM supposent que les expressions de gènes peuvent être discrétisées grâce à une échelle discrète finie. La validité de cette hypothèse de discrétisation est évidemment un critère déterminant dans la réussite de la méthode. La discrétisation a plusieurs avantages : d'abord, elle permet de rendre la méthode moins sensible au bruit ; ensuite, elle peut permettre de capturer facilement des liaisons non-linéaires. Mais une mauvaise discrétisation peut conduire à un non-sens biologique, et à une perte dramatique d'information. Pour ces raisons, ces méthodes n'ont pas été utilisées dans le cadre de cette thèse.

Réseaux Bayésiens et Réseaux Bayésiens Dynamiques Cette méthode a connue un large succès, notamment grâce à sa flexibilité importante. Les RB sont à la jonction de la théorie des graphes et de la théorie des probabilités (Rau 2011). Nous allons les définir maintenant de façon formelle.

Supposons tout d'abord que nous disposons d'un graphe $\mathcal{G} = \{V, E\}$ où V et E sont des variables aléatoires. Nous avons V qui représente les nœuds du graphe (vertice, en anglais), et E qui représente les arcs entre ces nœuds (edge, en anglais). Dans notre cas, les nœuds représentent les gènes, et les arcs les interactions entre ces gènes, ainsi qu'une famille de probabilités conditionnelles $\mathcal{F}$ paramétrisée par Θ (Husmeier 2005).

S'il existe un lien allant du nœud V_1 vers le nœuds V_2, alors V_1 sera considéré comme un parent pour le nœud V_2, et V_2 sera lui même considéré comme un descendant pour le nœud V_1. Dans un graphe avec N nœuds, nous noterons $Pa^{\mathcal{G}}(V_i)$ l'ensemble des parents pour le nœud V_i, $i = 1, ..., N$. Notons, que dans un cadre non-temporel, une tel définition interdit par essence l'existence de tout cycle dans le graphe.

On suppose ensuite que le graphe $\mathcal{G}$ respecte l'hypothèse de Markov, c'est-à-dire :

Hypothèse de Markov : *Chaque nœud est indépendant de tous les nœuds qui ne sont pas ses descendants, conditionnellement à ses parents.*

Par conséquent, on peut écrire :

$$\mathbb{P}(V_1, ..., V_n) = \prod_{i=1}^{N} \mathbb{P}\left(V_i | Pa^{\mathcal{G}}(V_i)\right).$$

Pour spécifier le RB, il faut ensuite spécifier $\mathbb{P}\left(V_i | Pa^{\mathcal{G}}(V_i)\right)$; cela peut se faire de deux manières, soit en utilisant un modèle multinomial (ce qui suppose, évidemment, que l'on ait d'abord discrétisé les variables) soit en utilisant un modèle gaussien. Dans le premier cas, il suffit alors de calculer la probabilité de chaque état selon l'état des parents (Friedman *et al.* 2000). L'algorithme BANJO (Yu *et al.* 2004), souvent utilisé, est basé sur cette méthode également.

L'hypothèse de Markov impose au graphe $\mathcal{G}$ d'être acyclique. C'est-à-dire qu'on ne peut pas avoir, par exemple : V_1 influence V_2, V_2 influence V_3, et V_3 influence V_1. C'est une des raisons pour lesquelles les RBD ont été développés (Murphy et Mian 1999). En effet, l'ajout de la dimension temporelle permet de lever l'hypothèse d'acyclicité.

Les modèles les plus utilisés dans ce cadre sont les modèles espaces état, dans lesquels on suppose qu'un processus non-observé intervient dans le modèle (Beal *et al.* 2005, Rau *et al.* 2010) ou dans une version non-linéaire (Quach *et al.* 2007).

Enfin, le temps de calcul des modèles bayésiens étant algorithmiquement long, l'incorporation d'information *a priori* permet d'obtenir des algorithmes plus rapides (Young *et al.* 2014).

1.6 Quelle utilité pour les réseaux biologiques

Nous venons de voir comment inférer un réseau de régulation génique. Ce réseau de régulation génique doit être vu comme une modélisation d'un système biologique complexe, en l'occurrence le programme de régulation génique. Nous avons vu dans le début de ce chapitre que la compréhension d'un système complexe passe par quatre étapes successives : description de la topologie, de la structure, puis détermination de la dynamique, puis utilisation de la connaissance de la structure et de la dynamique pour contrôler

le système, et enfin la modification orientée du système.

Le réseau de gènes doit être considéré comme étant la structure du système complexe étudié. Cependant, les méthodes d'inférence à équation, du fait de leur nature, permettent d'obtenir dans le même temps la dynamique du système. C'est pour cette raison que ces méthodes seront privilégiées dans cette thèse. En particulier, nous reviendrons dans le chapitre suivant sur un modèle de régression parcimonieuse classique : la régression Lasso.

Inférer un réseau de gènes permet donc de révéler la structure (et la dynamique selon la méthode choisie) du programme génique étudié. Ce qui est intéressant, c'est que les structures des réseaux de gènes, et plus généralement des réseaux métaboliques, montrent une topologie semblable. En effet, comme nous l'avons déjà vu, ces réseaux sont invariants d'échelle (Barabási et Albert 1999). Cette propriété a été vérifiée dans les réseaux de protéines à protéines, dans les réseaux de gènes, chez l'homme et chez la levure (Vidal *et al.* 2011, Barabási et Oltvai 2004, Seebacher et Gavin 2011). Pourtant, il faut distinguer les distributions des degrés entrants et sortants. En effet, la distribution des degrés sortants semblent bien suivre une distribution de type puissance, conduisant à avoir beaucoup de gènes faiblement régulateurs et quelques gènes fortement régulateurs. En revanche, la distribution des liens entrants, c'est-à-dire le nombre de gènes régulant un gène donné semble être de type exponentielle, indiquant que les gènes régulés par un grand nombre de régulateurs sont exponentiellement rares (Deplancke *et al.* 2006). Ceci conforte l'idée de parcimonie que nous avons déjà introduite. La raison pour laquelle l'organisation invariante d'échelle est retrouvée dans la plupart des réseaux de gènes semble être due au principe d'attachement préférentiel (Barabási et Albert 1999). Selon ce principe, le nombre de nœuds augmente petit à petit, et chaque nouveau nœud s'intègre dans le réseau déjà formé. Sa probabilité de liaison avec un gène déjà intégré est d'autant plus forte que ce gène est déjà fortement connecté. Parfois, ce phénomène est appelé "les riches deviennent encore plus riches". Il a été montré que, dans le cadre des réseaux biologiques, cette hypothèse est fortement probable (Pastor-Satorras *et al.* 2003).

Maintenant que nous avons montré qu'il existait une universalité des structures des réseaux biologiques, nous allons nous intéresser à l'interprétation biologique des éléments de ces structures. Les réseaux biologiques sont donc de type invariant d'échelle. Par conséquent, ils disposent de hubs, c'est-à-dire de nœuds fortement connectés. Une première question apparaît donc : que sait-on de ces nœuds ? quelle est leur importance biologique ?

Même si ce n'est pas exactement le sujet de cette thèse, regardons ce qui a été écrit sur les hubs dans les réseaux protéines à protéines (dont on attend qu'ils aient un lien important avec les réseaux de gènes) :

— ils correspondent à des gènes essentiels (Jeong *et al.* 2001),
— ils sont plus vieux et ont évolué plus lentement (Fraser *et al.* 2002),
— ils ont tendance à être plus abondants (Ivanic *et al.* 2009),
— leur suppression a des conséquences phénotypiques plus importantes

que les autres protéines moins connectées (Yu *et al.* 2008).

De manière plus générale, le nombre de connections semble être un indicateur pertinent de l'importance d'une protéine. Par exemple, chez les patients ayant un cancer, les protéines liées au cancer sont deux fois plus connectées que les autres (Jonsson et Bates 2006).

Cette structure invariante d'échelle a une autre conséquence : les délétions ou ruptures de fonctionnement d'un nœud sont peu importantes tant qu'elles ne touchent pas un hub. En revanche, dès lors qu'un hub est touché, un partitionnement du réseau est observé (Albert et Barabási 2002). La validité de cette conclusion pour les réseaux biologiques peut être vérifiée en associant la sévérité d'une inhibition de gènes avec le nombre d'interactions de ce gène. Plusieurs études montrent effectivement une forte corrélation entre cette association. Par exemple, 73% des gènes de S. cerevisiae ne sont pas importants, dans le sens où leur inhibition n'a pas d'effet phénotypique important (Giaever *et al.* 2002). Les réseaux sont donc robustes face à des ruptures de fonctionnement aléatoires. Par ailleurs, la probabilité qu'un gène soit essentiel (c'est-à-dire que son absence entraîne la mort de la cellule) est corrélée au nombre d'interactions de sa protéine (Jeong *et al.* 2001, Said *et al.* 2004). Cela indique que la cellule est en revanche très vulnérable à une perturbation de ses gènes hubs. La protéine p53, par exemple, est une protéine anti-tumeurs. Elle a été trouvée comme hub dans plusieurs études. Dans la moitié des tumeurs cette protéine est inactivée par mutation, ce qui confirme le lien entre hub et vulnérabilité cellulaire pour les hubs (Vogelstein *et al.* 2000).

Nous souhaitons encore rendre le lecteur attentif à une différence importante entre hubs : il y a d'une part les hubs constants (party hubs, en anglais) et les hubs de circonstances. Les premiers sont des hubs quels que soient les circonstances ou les moments d'étude tandis que les seconds ne revêtent leur caractère de hubs que dans des circonstances particulières (Seebacher et Gavin 2011, Han *et al.* 2004). Les hubs constants semblent caractériser des modules fonctionnels tandis que les hubs de circonstances serviraient plutôt à connecter les modules fonctionnels les uns aux autres (Han *et al.* 2004).

Si nous regardons plus en détail la structure des réseaux biologiques, plusieurs remarques peuvent encore être formulées :

— les gènes co-exprimés partagent souvent une même fonction,
— certains motifs, comme les boucles de régulation et les boucles de régulation inverses sont significativement plus présents (Balazsi *et al.* 2005, Shen-Orr *et al.* 2002) : ces motifs permettent d'apporter une réponse biologique rapide à une situation donnée,
— la probabilité qu'il existe deux chemins pour aller d'un gène à un autre dans le réseau est importante (Papin et Palsson 2004) : l'intérêt ici est évident en cas de rupture d'un des chemins.

L'étude des phénomènes biologiques sous l'angle de vue des systèmes

complexes amène notre compréhension du vivant vers une vision plus globale à l'échelle du système tout entier. Mais cela serait vain, si nous n'avions pas à la clef, l'espoir - sinon la promesse - de permettre à la médecine et à ses patients de trouver de nouvelles perspectives de traitement. Et mieux que cela encore, c'est la promesse d'une nouvelle vision de la médecine qui transparaît, celle dite des quatre P : Prédictive, Préventive, Participative et Personnalisée (Sobradillo *et al.* 2011, Hood *et al.* 2012).

1.7 De la biologie des systèmes à la médecine des systèmes

L'utilisation de la biologie des systèmes complexes dans les concepts en médecine, par le biais de procédures itératives entre données et modèles mathématiques et statistiques a pour nom "médecine des systèmes".

Si nous essayons de prendre un peu de recul, une question légitime se pose : en quoi la médecine des systèmes, en opposition à la médecine conventionnelle pratiquée aujourd'hui, va-t-elle être un plus pour le patient ? Pourquoi, comme le dit Charles Auffray, sommes-nous "véritablement à un moment historique de transition, équivalent à celui de la Renaissance qui a précédé l'émergence de la science moderne" [2] ?

Si de telles perspectives sont évoquées, c'est en partie parce que l'étude de la biologie en tant que système complexe du vivant a permis d'aboutir aux conclusions suivantes (Auffray *et al.* 2009, Wolkenhauer *et al.* 2013) :

— beaucoup de maladies trouvent leur origine dans un dysfonctionnement cellulaire, nécessitant une compréhension profonde des mécanismes inhérents au fonctionnement cellulaire,
— l'apparition des maladies est un processus non-linéaire, demandant une maîtrise des paramètres biologiques aux niveaux moléculaire, cellulaire et physiologique,
— les avancées technologiques permettent l'acquisition de jeu de données de mesure de très grandes dimensions, à des niveaux différents ; ces différentes acquisitions sont hétérogènes et leur étude simultanée reste un défi.

De nos lectures, nous comprenons que cette médecine ne pourra réussir que si elle parvient à devenir multi-échelle et si elle apporte (à terme) une modélisation spécifique à chacun des niveaux de cette échelle (Auffray et Nottale 2008, Nottale et Auffray 2008). Cette réussite dépendra également de nombreux autres éléments comme la construction de plans d'expérience adaptés, l'évolution des méthodes bio-informatiques (Clermont *et al.* 2009)...

Les jalons de la médecine des systèmes et de la médecine des quatre P sont donc posés. Et cette médecine n'est pas une médecine à long terme, mais bien une médecine destinée à éclore dans les toutes prochaines années. Par exemple, le dosage optimal de l'anticoagulant warfarine est directement

2. `http://www.millenaire3.com/uploads/tx_ressm3/Charles_Auffray_EISBM_.pdf`

lié à certaines mutations de gènes (Cooper *et al.* 2008, Takeuchi *et al.* 2009, Mitchell *et al.* 2011, Gaikwad *et al.* 2013, Giri *et al.* 2014).

1.8 Objectifs de cette thèse

Nous avons donc vu que des modifications dans le programme génique - modélisé dans notre cas par des réseaux de régulation de gènes -, peuvent avoir des conséquences phénotypiques importantes (voir Figure 1.12). En particulier, le cancer peut être l'une de ces conséquences. La question clef, celle qui a motivé tous nos efforts, est à la fois fort simple à énoncer, et fort difficile à résoudre :

Est-il possible de revenir en arrière ? Est-il possible de réorienter le programme génique en appliquant les perturbations inverses à celles qui ont conduit le patient à avoir un cancer ?

Dans les termes utilisés dans ce chapitre, notre question peut se traduire par les questions suivantes :

— est-il possible de décrire la structure du système, dans notre cas un programme cancéreux ?
— est-il possible de comprendre la dynamique de ce système, c'est-à-dire nous est-il possible de prédire les effets d'une perturbation de ce système ?
— finalement, est-il possible de moduler ce système, pour l'amener d'un état tumoral à un état sain ?

Nos travaux nous ont permis d'apporter des éléments de réponse pour les deux premières questions. Ces travaux sont présentés dans le corps principal de cette thèse, qui est l'objet de la partie II. La troisième question est plus délicate et nous n'avons à ce jour que quelques éléments à notre disposition. Cependant, nous développons dans le Chapitre 8 une méthodologie statistique, qui, améliorant la précision des méthodes de sélection de variables, doit être considérée comme une étape clef vers la modulation orientée du système. Par ailleurs, dans le Chapitre 9, nous détaillons la stratégie que nous avons mise en place dans ce sens. Des résultats biologiques tout à fait préliminaires semblent conforter l'approche que nous proposons.

Afin de pouvoir répondre à ces questions, un modèle biologique pertinent a été développé (voir Chapitre 3). Ce modèle biologique, basé sur l'étude d'un cancer, nous a conduit à proposer la conception de programme génique activé en cascade. Dans un tel programme, nous pouvons décrire l'impact d'une perturbation par une suite de réactions ordonnées temporellement. Cela nous a conduit logiquement à proposer une façon adaptée de modéliser ces programmes sous la forme de réseaux en cascade. Nous présentons cela en détail dans le Chapitre 3.

Avant cela, nous nous permettons de décrire plus en détail la régression parcimonieuse de type Lasso (Chapitre 2). L'objectif d'un tel chapitre est double : il s'agit d'abord de présenter un outil statistique qui sera largement

utilisé dans les chapitres suivants, mais aussi de présenter ses avantages et ses limites. Un tel aperçu doit permettre au lecteur de mieux comprendre notre travail, et justifie largement les efforts que nous déploierons au Chapitre 8, où nous proposerons un algorithme capable de lever certaines des limites du Lasso.

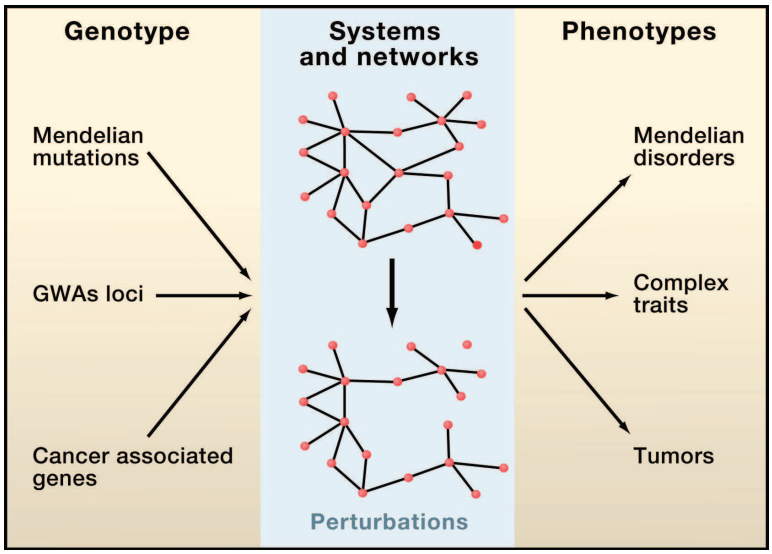


FIGURE 1.12 – *Impacts des perturbations dans les systèmes biologiques. Figure issue de l'article Villaverde* et al. *(2013).*

Régression Lasso

2

Ce chapitre permettra de faire quelques rappels sur la régression Lasso qui est une technique permettant de sélectionner les variables influentes dans une régression, quand bien même le nombre de ces variables serait supérieur au nombre d'observations. La régression Lasso sera en outre largement utilisée dans les chapitres suivants, et c'est pourquoi il nous a semblé essentiel de faire un état de l'art. Un état de l'art plus général sur les méthodes de sélection pourra être trouvé au Chapitre 8.

2.1 Introduction et historique

Supposons que dans l'étude d'un phénomène nous observons M variables pour N échantillons, et supposons, compte tenu des moyens de mesure actuels, que nous ayons M un nombre très grand, possiblement plus grand que N. Dans une régression classique, quand M est grand, il devient difficile de trouver les variables influentes, et les tests classiques donnent un nombre trop grand de faux positifs. De plus, quand M est plus grand que N, la régression classique n'est tout simplement plus possible. En effet, si nous notons $\boldsymbol{X}$ la matrice des observations, le cas $M > N$ entraine que ${}^t\boldsymbol{X}\boldsymbol{X}$ est une matrice de déterminant nul.

La régression linéaire avec pénalité Lasso (Tibshirani 1996) est une des manières de traiter ce problème. Cette méthode est apparue en 1996, et bien que très utilisée aujourd'hui, elle n'a pas eu un grand écho à sa sortie. Il y a deux raisons majeures à cela : tout d'abord, l'algorithme proposé pour résoudre le Lasso était coûteux en termes de temps de calcul, et ensuite l'informatique n'étant pas encore aussi développée qu'aujourd'hui, les cas avec un très grand M étaient rares. Cependant en 2004 est apparu l'algorithme LARS-Lasso (Efron *et al.* 2004) qui a permis de résoudre le problème lasso avec la même complexité qu'une régression linéaire simple. Cet article a marqué le début de l'intérêt pour cette technique, et il est le point de départ de son expansion.

Revenons cependant en 1996, alors que vient d'être publié l'article sur la régression Lasso (Tibshirani 1996) : qu'en est-il du problème de la sélection de variables ? En grande partie sont utilisées des méthodes de type "Stepwise regression", ou encore "Régression pas à pas". L'idée ici est de partir d'un modèle donné (le plus souvent le modèle complet ou le modèle vide), et de modifier à chaque étape le modèle de régression considéré, en enlevant une

ou plusieurs variables de ce modèle, ou au contraire en en retirant une ou plusieurs. Voici un exemple d'une telle procédure :

1. Choisir la variable x_1 la plus adéquate au modèle (en fonction d'un critère défini à l'avance, comme le critère AIC, BIC,...)
2. Pour i de 2 à M :
   (a) Calculer ce même critère pour tous les modèles possibles à i variables contenant $x_1, ..., x_{i-1}$. Il y en a donc $M - i$.
   (b) Choisir le meilleur modèle entre les $M - i$ modèles ci-dessus, et le modèle à l'étape précédente :
   — Si le meilleur modèle est le modèle de l'étape précédente, nous arrêtons la boucle,
   — Si le meilleur modèle est un des $M - i$ nouveaux modèles, nous continuons la boucle (*i.e.* $i = i + 1$).

L'inconvénient de telles méthodes est évident : plus le nombre de variables M est grand, et plus l'algorithme sera long à converger (*N.B.* : par exemple, dans l'algorithme ci-dessous, si $i = I$ il faudra faire $I + (I-1) + ... + (I-i) = \frac{(2I-i)(i+1)}{2}$ régressions).

L'année précédant la publication de la méthode du Lasso, en 1995, Breiman proposa une méthode de sélection de variables (Breiman 1995) dont Tibshirani expliquera s'en être largement inspiré dans une rétrospective parue en 2011 (Tibshirani 2011). L'idée de Breiman est de calculer l'estimateur des moindres carrés ordinaire (on note donc que cette méthode ne permet pas de traiter le cas $M > N$), avant de pénaliser, dans un deuxième temps distinct, ces cœfficients en rajoutant dans le modèle de régression une pondération.

Avant de continuer, et de présenter plus précisément la méthode de Breiman appelée "non negative garrotte", il faut poser quelques notations, utiles pour formaliser le problème. Nous avons déjà M le nombre de variables et N le nombre d'observations. Prenons pour convention que toute matrice ou vecteur sera noté en gras, tandis que tout réel, cœfficient de n'importe quel ordre, sera noté de façon normale. Autant que faire se peut, les majuscules seront réservées aux matrices, et les minuscules aux vecteurs. La notation ".", indicée à une matrice, signifie que nous considérons toutes les possibilités à la place de ".". Ainsi $\boldsymbol{X}$ représente une matrice, $\boldsymbol{x}_{i.}$ représente le vecteur formé de la ième ligne de la matrice $\boldsymbol{X}$, et x_{ij} représente l'élément situé à la ième ligne et à la jème colonne de $\boldsymbol{X}$.

Le problème considéré ici est d'expliquer les variations d'une variable réponse $\boldsymbol{y}$ (vecteur de dimension N) à partir de M vecteurs de même dimensions, assemblés en colonne dans une matrice $\boldsymbol{X}$. En supposant ce lien linéaire, nous pouvons établir le modèle classique de la régression linéaire multiple :

$$y_n = \sum_{m=1}^{M} \beta_m x_{nm} + \varepsilon_n, \quad \forall n \in 1, ..., N,$$

où $\boldsymbol{\varepsilon}$ est un vecteur d'erreurs gaussiennes i.i.d. centré et de même variance. La matrice $\boldsymbol{X}$ contient donc N lignes pour M colonnes, et l'élément x_{nm} correspond à l'observation faite pour le nième échantillon de la mième variable. L'estimation des paramètres dans la régression classique s'écrit alors comme un problème de minimisation :

$$\hat{\boldsymbol{\beta}}^{MCO} = \underset{\boldsymbol{\beta}\in\mathbb{R}^M}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left(y_n - \sum_{m=1}^{M} \beta_m x_{nm} \right)^2 \right]. \tag{2.1}$$

Comme nous l'avons déjà annoncé, l'estimateur de Breiman est construit en deux étapes, dont la première consiste à calculer l'estimateur des moindres carrés ordinaire obtenu par l'équation (2.1). Une pénalité est ajoutée dans le but de contraindre certains cœfficients de la régression à être nul. L'estimateur (noté NNG pour "non negative garrotte')' est solution du problème de minimisation suivant :

$$\hat{\boldsymbol{\beta}}^{NNG} = \underset{\boldsymbol{c}\in\mathbb{R}^M \quad s.c. \quad \|\boldsymbol{c}\|_1\leqslant\lambda}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left(y_n - \sum_{m=1}^{M} c_m \hat{\beta}_m^{MCO} x_{nm} \right)^2 \right] * \hat{\boldsymbol{\beta}}^{MCO}, \tag{2.2}$$

où $\lambda \in \mathbb{R}^+$ est la contrainte choisie par l'utilisateur, dont la valeur permet de moduler le niveau de parcimonie, et $*$ est le produit terme à terme. Maintenant, pour trouver l'estimateur Lasso (Tibshirani 1996) il suffit d'unifier ces deux étapes en une étape unique. La manière qui nous semble la plus naturelle, c'est-à-dire de porter directement la contrainte sur les cœfficients de la régression, est la bonne :

$$\hat{\boldsymbol{\beta}}^{Lasso\_1}(\lambda_1) = \underset{\boldsymbol{\beta}\in\mathbb{R}^M \quad s.c. \quad \|\boldsymbol{\beta}\|_1\leqslant\lambda_1}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left(y_n - \sum_{m=1}^{M} \beta_m x_{nm} \right)^2 \right], \tag{2.3}$$

où $\lambda_1 \in \mathbb{R}^+$ joue exactement le même rôle que dans l'estimateur de Breiman. Nous pouvons nous arrêter un instant pour faire quelques remarques faciles à observer mais qui permettent de bien comprendre comment fonctionne le Lasso :

- lorsque λ vaut zéro, la solution au problème Lasso est simplement le vecteur nul,
- lorsque λ devient "très grand", la solution au problème Lasso est exactement la solution des moindres carrés ordinaires,
- pour les valeurs intermédiaires de λ, en partant de zéro, nous obtenons des solutions de moins en moins parcimonieuses.

Pour bien comprendre comprendre le fonctionnement du Lasso, il peut être utile de s'intéresser à une méthode proche, dont l'étude des différences se révélera instructive ; si nous reprenons la formulation du Lasso dans l'équation (2.3) et que nous changeons la contrainte $\mathcal{L}_1$ en une contrainte $\mathcal{L}_2$, nous obtenons la régression Ridge :

$$\hat{\beta}^{Ridge} = \underset{\beta \in \mathbb{R}^M \quad s.c. \quad \|\beta\|_2^2 \leqslant \lambda}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left(y_n - \sum_{m=1}^{M} \beta_m x_{nm} \right)^2 \right]. \tag{2.4}$$

Pour bien comprendre ce qui se passe, il est utile de regarde la forme de ces deux pénalités (voir Figure 2.1).

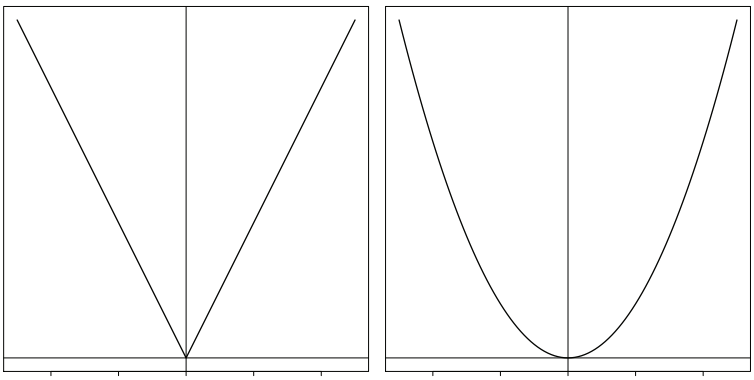


FIGURE 2.1 – *La fonction de pénalité du Lasso (à gauche) et celle du Ridge (à droite).*

Cependant, l'intérêt majeur vient en regardant les dérivées de ces deux fonctions. En effet, les dérivées représentent en quelque sorte le gain (en terme de diminution de la pénalité) que nous obtenons en diminuant les cœfficients de régression (voir Figure 2.2).

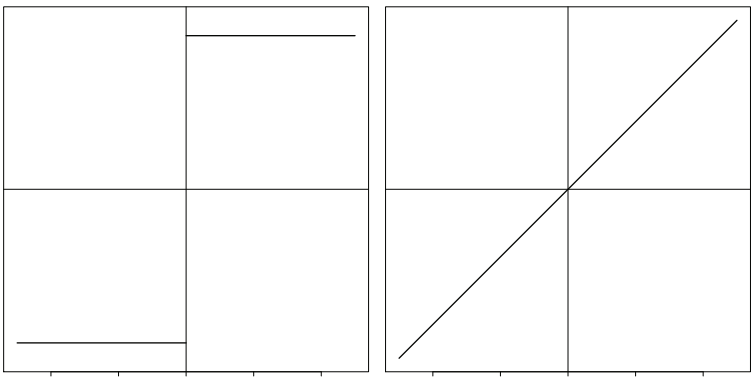


FIGURE 2.2 – *La fonction de pénalité dérivée du Lasso (à gauche) et celle du Ridge (à droite).*

Nous voyons donc que plus nous nous rapprochons de zéro, et plus le gain est faible dans la régression Ridge, alors qu'il est constant pour la régression Lasso : cela est une explication heuristique que nous pourrions donner sous la forme "la sélection de variables se fait par une non dérivabilité en zéro de la fonction de pénalité".

Il y a peut-être une façon encore plus agréable de considérer ces deux régressions. Supposons que nous disposions de l'estimation des moindres carrés ordinaire, et que nous tracions progressivement les ellipsoïdes sur lesquelles les erreurs au carré restent constantes. Un même exemple, dans un modèle

à deux covariables, est montré dans les Figures 2.3 et 2.4. Ces figures permettent de voir comment les ellipsoïdes autour de l'estimateur des moindres carrés vont intersecter la boule unité (au sens de la norme $\mathcal{L}_1$ pour le Lasso, au sens de la norme $\mathcal{L}_2$ pour la régression Ridge), et comment, la différence de topologie entre ces deux boules, va permettre (ou non) de reléguer un des deux cœfficients à zéro, permettant ainsi de faire de la sélection de variables.

Le manque de régularité en zéro de la norme $\mathcal{L}_1$ se traduit, dans la topologie de sa boule unité, par des angles. La boule unité $\mathcal{L}_2$ est parfaitement lisse.

2.2 Premières propriétés de la régression Lasso

Aussi plaisant soit-il de considérer le Lasso selon la formulation de l'équation (2.3), il est souvent utile, en pratique, de considérer la version ci-dessous, par ailleurs plus courante :

$$\hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2) = \underset{\boldsymbol{\beta}\in\mathbb{R}^M}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left(y_n - \sum_{m=1}^{M} \beta_m x_{nm} \right)^2 + \lambda_2 \sum_{m=1}^{M} |\beta_m| \right], \quad (2.5)$$

où $\lambda_2 \in \mathbb{R}^+$ est un paramètre de contrôle de la parcimonie du modèle. Si nous avons pris soin de noter $\hat{\boldsymbol{\beta}}^{Lasso\_1}$ et $\hat{\boldsymbol{\beta}}^{Lasso\_2}$, c'est que les deux formulations du problème Lasso ne sont pas, *stricto sensu*, équivalentes. Pour s'en convaincre, il suffit de considérer les deux cas extrêmes :

- lorsque $\lambda_1 = \lambda_2 = 0$, il est facile de voir que $\hat{\boldsymbol{\beta}}^{Lasso\_1}(\lambda_1)$ est le vecteur nul tandis que $\hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2)$ est la solution des moindres carrés ordinaires,
- lorsque $\lambda_1 = \lambda_2 = +\infty$, il est facile de voir que l'inverse se produit : $\hat{\boldsymbol{\beta}}^{Lasso\_1}(\lambda_1)$ est alors la solution des moindres carrés ordinaires tandis que $\hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2)$ devient le vecteur nul.

Intuitivement, il est aisé de comprendre que les deux paramètres de pénalisations des deux écritures du Lasso agissent de façon contraire. Cependant, les deux définitions sont équivalentes dans un certain sens. Plus précisément, nous avons le lemme suivant :

Lemme 2.1 *Supposons que la résolution du Lasso dans sa seconde écriture (2.5) donne pour solution $\hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2)$, pour un certain $\lambda_2 \in \mathbb{R}^+$ choisi par l'utilisateur. Alors il existe un $\lambda_1 \in \mathbb{R}^+$ tel que la solution obtenue par la première formulation (2.3), $\hat{\boldsymbol{\beta}}^{Lasso\_1}(\lambda_1)$, est telle que :*

$$\hat{\boldsymbol{\beta}}^{Lasso\_1}(\lambda_1) = \hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2).$$

Démonstration. Posons $\lambda_1 = \sum_{m=1}^{M} |\hat{\beta}_m^{Lasso\_2}(\lambda_2)|$ et montrons alors que $\hat{\boldsymbol{\beta}}^{Lasso\_1}(\lambda_1) = \hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2)$.

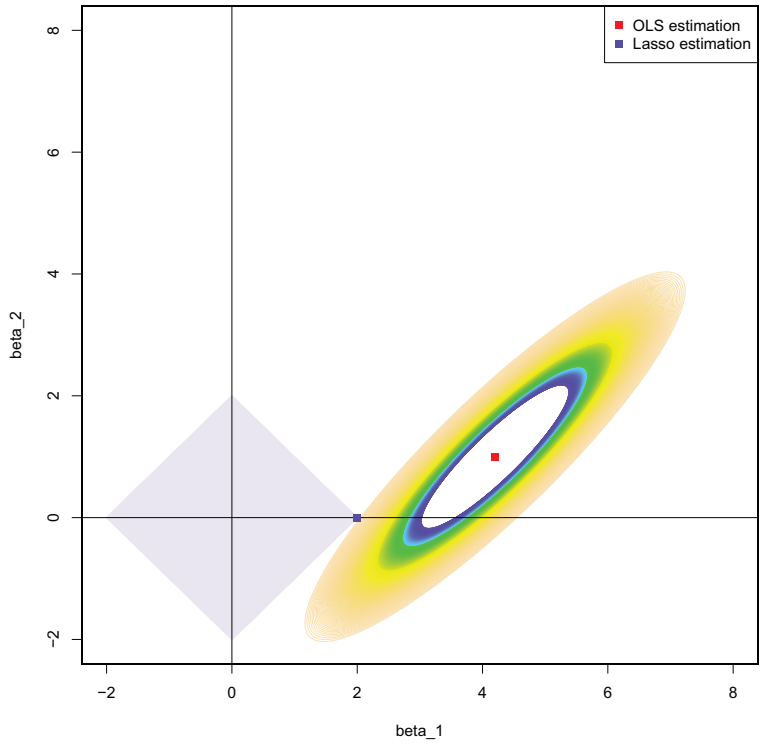


FIGURE 2.3 – *La boule unité $\mathcal{L}_1$, l'estimation Lasso obtenue à partir de l'estimation MCO.*

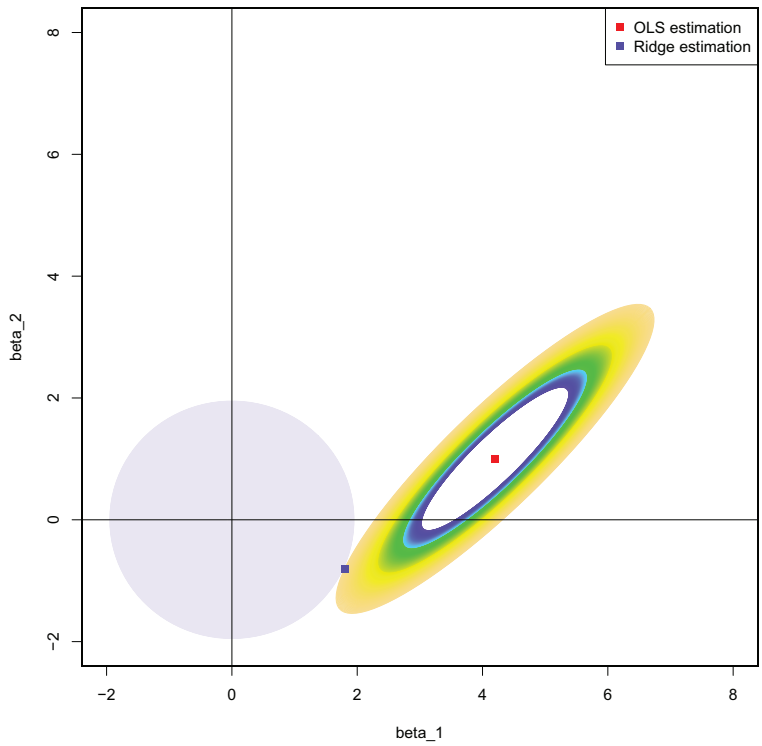


FIGURE 2.4 – *La boule unité $\mathcal{L}_2$, l'estimation Ridge obtenue à partir de l'estimation MCO.*

Pour tout $\boldsymbol{\beta} \in \mathbb{R}^M$ nous avons :

$$\|\boldsymbol{\beta}\|_1 \leqslant \lambda_1 \Rightarrow \lambda_2\|\boldsymbol{\beta}\|_1 \leqslant \lambda_2\|\hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2)\|_1. \tag{2.6}$$

L'expression dans l'équation (2.5) résultant du minimum de la somme de deux termes positifs, l'inégalité ci-dessus (2.6) implique :

$$\sum_{n=1}^{N}\left(y_n - \sum_{m=1}^{M}\beta_m x_{nm}\right)^2 \geqslant \sum_{n=1}^{N}\left(y_n - \sum_{m=1}^{M}\hat{\beta}_m^{Lasso\_2}(\lambda_2)x_{nm}\right)^2,$$

et par voie de conséquence, le minimum de :

$$\sum_{n=1}^{N}\left(y_n - \sum_{m=1}^{M}\beta_m x_{nm}\right)^2$$

pour tout $\boldsymbol{\beta} \in \mathbb{R}^M$ tel que $\|\boldsymbol{\beta}\|_1 \leqslant \lambda_1$ est $\hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2)$. □

Cette seconde écriture permet d'avoir une autre vision du Lasso ; celle-ci est présentée dans la remarque ci-dessous :

Remarque 14 *Supposons que $\hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2)$ est la solution du problème Lasso dans sa deuxième écriture. Supposons de plus que toutes les covariables sont orthogonales deux à deux, et appelons, $\hat{\boldsymbol{\beta}}^{MCO}$ la solution obtenue par moindres carrés ordinaires. Alors :*

$$\hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2) = sign(\hat{\boldsymbol{\beta}}^{MCO}) * \max(0, |\hat{\boldsymbol{\beta}}^{MCO}| - \frac{\lambda_2}{2}).$$

Autrement dit, tout en gardant le signe des moindres carrés ordinaires, l'estimateur de la régression Lasso peut ici être vu comme une simple translation de l'estimateur des moindres carrés ordinaires tronqué en zéro.

Sans rentrer dans les détails, cette formulation permet de relever un des problèmes du Lasso : son biais. En effet, tandis qu'il peut sembler judicieux de translater les "petits cœfficients" vers 0, il est certain que ce traitement, appliqué à toutes les variables, même les plus importantes, amène un biais.

Une dernière façon de voir le Lasso est de considérer le problème sous un angle bayésien (Tibshirani 1996) :

Remarque 15 *La régression Lasso peut-être considérée d'un point de vue bayésien où une loi de Laplace centrée en 0 serait utilisée comme a priori sur les cœfficients.*

Reste maintenant à trouver une façon efficace d'estimer les cœfficients de cette régression que pénalise une norme $\mathcal{L}_1$. Dans l'article originel du Lasso (Tibshirani 1996) l'auteur présente plusieurs stratégies, toutes trop lentes. Comme nous l'avons déjà dit, c'est l'algorithme LARS (Efron *et al.* 2004) qui appliqué au Lasso permettra d'obtenir, avec la complexité d'une régression simple, le chemin entier des solutions, nommé chemin de régularisation (*i.e.* non pas à λ fixé, mais pour $\lambda \in \mathbb{R}^+$) ! Comme nous le verrons, cela est possible parce que ce chemin est linéaire par morceaux. Avant de construire la solution Lasso par l'algorithme LARS-Lasso, nous allons effectuer quelques rappels succincts sur les conditions de Karush-Kuhn-Tucker,

utiles pour la suite de notre raisonnement. Ces conditions sont le pendant des multiplicateurs de Lagrange pour les maximisations sous des contraintes d'inégalités.

Remarque 16 *Dans la deuxième formulation du Lasso, il est facile de voir que si ${}^t\boldsymbol{X}\boldsymbol{X}$ est une matrice symétrique définie positive, alors le problème Lasso admet une unique solution. En effet, le problème de minimisation possède alors une fonction objectif composée de la somme d'une fonction strictement convexe et d'une fonction convexe.*

Avant d'en venir à ces détails techniques, nous aimerions rendre attentif le lecteur au choix de la pénalité $\mathcal{L}_1$. C'est l'objet de la section suivante.

2.3 Pourquoi la norme $\mathcal{L}_1$?

Bien que justifiée par cette approche historique, l'utilisation d'une régression sous une contrainte $\mathcal{L}_1$ peut interpeller le lecteur dans un premier temps. En effet, comme nous l'avons montré, la régression Lasso permet d'obtenir une estimation parcimonieuse du vecteur des cœfficients. Le lecteur est alors en droit de se demander pourquoi la contrainte (ou la pénalisation, selon l'écriture), au lieu de porter sur la norme $\mathcal{L}_1$ du vecteur des cœfficients, ne porte pas sur le nombre de cœfficients non nuls de ce vecteur. Autrement dit, pourquoi avoir choisi la norme $\mathcal{L}_1$ au lieu de la "pseudo" norme $\mathcal{L}_0$, définie de la façon suivante :

$$\|\boldsymbol{x}\|_0 = \sum_{n=1}^{N} I_{x_n \neq 0}, \ \forall \boldsymbol{x} \in \mathbb{R}^N.$$

Nous rendons le lecteur attentif au fait que ceci n'est pas une norme. En effet, nous voyons facilement que :

$$\|\boldsymbol{\lambda} x\|_0 = \|\boldsymbol{x}\|_0, \ \forall \lambda \in \mathbb{R}*.$$

Comme nous le montrons dans le Chapitre 8, la pseudo norme $\mathcal{L}_0$ est largement utilisée en statistique. Elle s'appelle critère AIC, critère BIC, RIC, elle s'appelle le C_p de Mallow... L'utilisation de ce genre de critère doit être souvent combiné avec des stratégies de recherche efficaces, comme la procédure de régression ascendante que nous avons décrite dans le début de ce chapitre. La difficulté de ces méthodes réside dans la nécessité sous-jacente d'une puissance de calcul trop souvent rédhibitoire. En effet, le nombre de modèles à examiner est trop important lorsque les données sont de grande dimension. De manière tout à fait générale, le problème de minimisation suivant :

$$\underset{\boldsymbol{\beta} \in \mathbb{R}^M}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left(y_n - \sum_{m=1}^{M} \beta_m x_{nm} \right)^2 + \lambda \|\boldsymbol{\beta}\|_0 \right],$$

avec λ un paramètre défini par l'utilisateur est un problème NP difficile.

L'utilisation de la norme $\mathcal{L}_1$ peut alors être vu comme un succédané à la norme $\mathcal{L}_0$. En terme mathématique, nous parlons de relaxation. A ce moment, deux questions peuvent apparaître comme pertinentes. La première

est celle qui consiste à se demander si cette relaxation aboutit au même résultat et, si oui, dans quelles conditions. De manière générale, il est évident que l'on aboutit pas aux mêmes estimations. Cependant, il a été démontré (dans différents cas, avec ou sans présence de terme d'erreur...) que des conditions existent sous lesquelles les résultats des régressions $\mathcal{L}_1$ et $\mathcal{L}_0$ pénalisées aboutissent aux mêmes résultats (Donoho et Elad 2003, Candes *et al.* 2006)... La deuxième question légitime est celle que tout statisticien se pose lorsqu'il définit un estimateur : étant donné mon modèle, soumis aux hypothèses que je lui impose, mon estimateur est-il capable de "bien" estimer le paramètre cible ? Dans la suite, c'est ce point de vue que nous adopterons, en définissant le modèle, les hypothèses, et ce que nous pouvons attendre d'un "bon" estimateur.

Ce point étant résolu, nous reprenons la suite de notre raisonnement et définissons les conditions de Karush-Kuhn-Tucker qui vont nous permettre de résoudre le problème de l'estimation des paramètres de la régression Lasso.

2.4 Conditions de Karush-Kuhn-Tucker

Soit $f : \mathbb{R}^M \to \mathbb{R}$ une fonction à plusieurs variables, convexe et continûment différentiable, et intéressons-nous à la recherche d'un minimum pour f. Dans le cas général, la convexité de f nous assure l'existence d'un minimum. Ce minimum peut être trouvé soit analytiquement, soit par une solution approchée issue d'un algorithme itératif. Dans certains cas, il est d'intérêt de chercher le minimum non pas sur $\mathbb{R}^M$ tout entier, mais sur un sous-ensemble. Ce sous-ensemble est souvent décrit par une ou plusieurs contraintes sur la solution. Quand ces contraintes sont des égalités, c'est-à-dire lorsqu'il existe une fonction : $\mathbb{R}^M \to \mathbb{R}^K$, où K est le nombre de contraintes, le problème se réécrit alors :

$$\boldsymbol{x}^* \in \underset{\substack{\boldsymbol{x}\in\mathbb{R}^M \\ g(\boldsymbol{x})=\mathbf{0}_K}}{\operatorname{argmin}} \big[f(\boldsymbol{x})\big] \cdot$$

Il est bien connu que la solution à ce problème se trouve par la méthode des extrama liés (Gourdon 2000). Mais qu'en est-il si la contrainte est posée sous la forme d'une inégalité ? Supposons g convexe, et le problème se réécrit alors :

$$\boldsymbol{x}^* \in \underset{\substack{\boldsymbol{x}\in\mathbb{R}^M \\ g(\boldsymbol{x})\leqslant\mathbf{0}_K}}{\operatorname{argmin}} \big[f(\boldsymbol{x})\big] \cdot$$

L'extension naturelle des extrema liés est les conditions KKT, pour Karush-Kuhn-Tucker (Karush 1939). La première étape reste la même que dans les extrema liés ; il faut calculer le Lagrangien, défini par :

$$L(\boldsymbol{\gamma}, \boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{M} \gamma_i g_i(\boldsymbol{x}) \cdot$$

C'est alors qu'interviennent les conditions KKT :

Définition 17 (Conditions KKT) *Nous appelons conditions KKT l'ensemble des quatre conditions suivantes :*

$$\begin{cases} \gamma_k \in \mathbb{R}^+, \quad k = 1, ..., K. \\ g(\boldsymbol{x}^*) \leqslant 0 \\ \nabla_{\boldsymbol{x}} L(\gamma, \boldsymbol{x}^*) = 0 \\ g_k(\boldsymbol{x}^*)\gamma_k = 0, \quad k = 1, ..., K. \end{cases} \tag{2.7}$$

La condition fondamentale (par rapport aux extrema liés) est la quatrième, dont nous verrons qu'elle aura une importance de première ordre dans la résolution Lasso par la méthode LARS-Lasso. Cette dernière stipule que, si une condition k n'est pas saturée (*i.e.* $g_k(\boldsymbol{x}^*) \neq 0$), alors forcément $\gamma_k = 0$.

Le théorème, que nous donnons ici sans démonstration, permet de résoudre le problème de minimisation avec des contraintes sous forme d'inégalités :

Théorème 18 *Soit (P) le problème de la recherche d'un $\boldsymbol{x}^*$ tel que :*

$$\boldsymbol{x}^* \in \underset{\substack{\boldsymbol{x} \in \mathbb{R}^M \\ g(\boldsymbol{x}) \leqslant \boldsymbol{0}_K}}{\operatorname{argmin}} \big[f(\boldsymbol{x}) \big],$$

avec f et g deux fonctions convexes comme décrites précédemment, et f continûment différentiable. Le problème (P) admet une solution si et seulement si il existe $\boldsymbol{\gamma} \in \mathbb{R}^K$ tel que les conditions KKT soient remplies.

Ce théorème, utile pour résoudre le problème dans sa première formulation (2.3), peut s'écrire de manière équivalente, pour la deuxième formulation (2.5), en annulant le gradient de :

$$f(\boldsymbol{\beta}) = \sum_{n=1}^{N} \left(y_n - \sum_{m=1}^{M} \beta_m x_{nm} \right)^2 + \lambda_2 \sum_{m=1}^{M} |\beta_m| \cdot$$

Une remarque pour clore cette section :

Remarque 19 *Sauf dans le cas où la matrice ${}^t\boldsymbol{X}\boldsymbol{X}$ est symétrique définie positive, l'existence d'une solution unique n'est pas assurée, puisque le problème n'est plus strictement convexe.*

2.5 L'algorithme LARS-Lasso

À partir de maintenant, et sauf mention contraire, le problème Lasso référera toujours à la deuxième écriture (2.5). Ainsi nous écrirons $\hat{\boldsymbol{\beta}}^L(\lambda)$ à la place de $\hat{\boldsymbol{\beta}}^{Lasso\_2}(\lambda_2)$.

D'autre part, nous supposerons que les régresseurs, ainsi que la variable réponse, sont centrés et réduits.

En utilisant les résultats de la partie précédente, nous obtenons les conditions d'optimalité suivantes :

Théorème 20 (Conditions d'optimalité) *Le vecteur $\hat{\boldsymbol{\beta}}^L(\lambda)$ est solution du problème Lasso dans sa deuxième écriture (2.5) si et seulement si :*

$$\begin{cases} {}^t\boldsymbol{X}\big(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^L(\lambda)\big) = \lambda\boldsymbol{\gamma} \\ \\ \forall i = 1, ..., p \ : \ \gamma_i \begin{cases} sgn(\hat{\beta}_i^L(\lambda)), & quand \quad \hat{\beta}_i^L(\lambda) \neq 0 \\ \in [-1;1] & quand \quad \hat{\beta}_i^L(\lambda) = 0. \end{cases} \end{cases} \tag{2.8}$$

Il est facile de voir que nous retrouvons la régression ordinaire lorsque $\lambda = 0$. En effet, si $\lambda = 0$ dans l'équation (2.8), la condition d'optimalité se résume à :

$$ {}^t\boldsymbol{X}\big(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^L(0)\big) = 0.$$

Autrement dit, $\boldsymbol{\beta}^L(0)$ est solution du problème Lasso si et seulement si les résidus sont orthogonaux à l'ensemble des prédicteurs, ce qui est exactement une condition nécessaire et suffisante dans la méthode des moindres carrés ordinaires.

Un examen approfondi du théorème 20 permet de comprendre comment le Lasso fonctionne. Pour cela, remarquons que ce théorème implique, entre autre, pour tout $i = 1, ..., M$:

$$\begin{aligned} & \hat{\beta}_i^L(\lambda) \neq 0 \Rightarrow \left|{}^t\boldsymbol{x}_i\big(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^L(\lambda)\big)\right| = \lambda \\ \Leftrightarrow \quad & \hat{\beta}_i^L(\lambda) \neq 0 \Rightarrow |< \boldsymbol{x}_i, \boldsymbol{\varepsilon} >| = \lambda, \end{aligned} \tag{2.9}$$

en notant $\boldsymbol{\varepsilon}$ les résidus de la régression :

$$\boldsymbol{\varepsilon} = \left|\big(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^L(\lambda)\big)\right|.$$

Notons que dans le cadre présent de variables réduites, le cœfficient de parcimonie λ peut être associé à un facteur de corrélation maximale. Ainsi, si $\lambda > 1$, le modèle choisi est le modèle nul, où le cœfficient de tous les régresseurs est nul (i.e., $\boldsymbol{\beta} = \boldsymbol{0}_M$).

Heuristiquement, les moindres carrés ordinaires "prennent" toute l'information des prédicteurs susceptibles d'expliquer les variations de la variable réponse. Dans la régression Lasso, seule une partie, définie par le cœfficient λ est prise en compte. Par conséquent, les covariables ne permettant d'expliquer qu'une faible partie de la variable réponse sont ignorées.

L'algorithme LARS-Lasso se dessine alors tout naturellement. Définissons tout d'abord l'ensemble des prédicteurs dont la corrélation linéaire avec les résidus est maximale (*i.e.*, dont la corrélation vaut λ) :

$$\Delta = \left\{ i \in 1, ..., M \ \ t.q. \ \ \left|{}^t\boldsymbol{x}_i\big(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^L(\lambda)\big)\right| = \lambda \right\}. \tag{2.10}$$

Alternativement, cet ensemble décrit le support de la régression :

Remarque 21 *Il est facile de voir que :*

— *Si* $i \in \Delta$ *alors* $\hat{\boldsymbol{\beta}}_i^L(\lambda) \neq 0$.
— *Si* $i \notin \Delta$ *alors* $\hat{\boldsymbol{\beta}}_i^L(\lambda) = 0$.

Cet ensemble permet aussi de définir la sous-matrice de $\boldsymbol{X}$ composée des régresseurs inclus dans le support :

Définition 22 *Notons* $\boldsymbol{X}_\Delta$ *la sous-matrice de* $\boldsymbol{X}$ *où nous avons sélectionné les colonnes correspondant aux indices dans* Δ.

Etant donné que nous souhaitons travailler dans les cas où $M >> N$, se pose le problème de l'inversibilité de la matrice ${}^t\boldsymbol{X}\boldsymbol{X}$; c'est pourquoi nous allons devoir nous satisfaire d'une notion un peu plus faible, dont la définition est donnée ci-dessous :

Définition 23 (Pseudo-inverse) *Soit* $\boldsymbol{A}$ *une matrice réelle. Une matrice* $\boldsymbol{A}^+$ *est appelée pseudo-inverse de* $\boldsymbol{A}$ *si :*

$$\begin{cases} \boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A} \\ \boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{A}^+ \\ {}^t\left(\boldsymbol{A}\boldsymbol{A}^+\right) = \boldsymbol{A}\boldsymbol{A}^+ \\ {}^t\left(\boldsymbol{A}^+\boldsymbol{A}\right) = \boldsymbol{A}^+\boldsymbol{A} \end{cases} \tag{2.11}$$

Propriété 24 *Dans le cas des matrices réelles :*

— *Le pseudo-inverse d'une matrice nulle est sa transposée.*
— *Le pseudo-inverse peut être vu comme limite :*

$$\lim_{\delta \to 0} \left({}^t\boldsymbol{A}\boldsymbol{A} + \delta \boldsymbol{I}\boldsymbol{d}\right)^{-1} {}^t\boldsymbol{A}.$$

— *Dans le cas où la matrice est carrée et non singulière, la définition de pseudo-inverse coïncide avec la notion d'inverse.*
— *Le pseudo-inverse, lorsqu'il existe, est unique.*

En utilisant les conditions d'optimalité de la propriété 2.8 (réécrit dans l'équation (2.9)), et l'ensemble Δ de la définition 22 :

$${}^t\boldsymbol{X}_\Delta\left(\boldsymbol{y} - \boldsymbol{X}_\Delta \hat{\boldsymbol{\beta}}_\Delta^L(\lambda)\right) = \lambda \boldsymbol{\gamma}_\Delta. \tag{2.12}$$

Dans l'équation (2.12) ci-dessus, le vecteur $\lambda\boldsymbol{\gamma}_\Delta$ est dans l'image de ${}^t\boldsymbol{X}_\Delta$. Par conséquent :

$$\begin{aligned} {}^t\boldsymbol{X}_\Delta {}^t\boldsymbol{X}_\Delta^+ \lambda\boldsymbol{\gamma}_\Delta &= {}^t\boldsymbol{X}_\Delta {}^t\boldsymbol{X}_\Delta^+ {}^t\boldsymbol{X}_\Delta\left(\boldsymbol{y} - \boldsymbol{X}_\Delta \hat{\boldsymbol{\beta}}_\Delta^L(\lambda)\right) \\ &\overset{def 23}{=} {}^t\boldsymbol{X}_\Delta\left(\boldsymbol{y} - \boldsymbol{X}_\Delta \hat{\boldsymbol{\beta}}_\Delta^L(\lambda)\right) \\ &= \lambda\boldsymbol{\gamma}_\Delta. \end{aligned}$$

Autrement dit, la propriété démontrée ci-dessus indique qu'une matrice multipliée par son pseudo-inverse agit comme l'identité lorsque que le vecteur appartient à l'image de ladite matrice.

En reprenant, à partir de l'équation (2.12) :

$$
\begin{aligned}
& {}^t\boldsymbol{X}_\Delta\big(\boldsymbol{y} - \boldsymbol{X}_\Delta\hat{\boldsymbol{\beta}}^L_\Delta(\lambda)\big) = \lambda\boldsymbol{\gamma}_\Delta \\
\Leftrightarrow \quad & {}^t\boldsymbol{X}_\Delta\boldsymbol{X}_\Delta\hat{\boldsymbol{\beta}}^L_\Delta(\lambda) = {}^t\boldsymbol{X}_\Delta\boldsymbol{y} - {}^t\boldsymbol{X}_\Delta{}^t\boldsymbol{X}^+_\Delta\lambda\boldsymbol{\gamma}_\Delta \\
\Leftrightarrow \quad & {}^t\boldsymbol{X}_\Delta\boldsymbol{X}_\Delta\hat{\boldsymbol{\beta}}^L_\Delta(\lambda) = {}^t\boldsymbol{X}_\Delta\big(\boldsymbol{y} - {}^t\boldsymbol{X}^+_\Delta\lambda\boldsymbol{\gamma}_\Delta\big) \\
\Leftrightarrow \quad & {}^t\boldsymbol{X}^+_\Delta{}^t\boldsymbol{X}_\Delta\boldsymbol{X}_\Delta\hat{\boldsymbol{\beta}}^L_\Delta(\lambda) = {}^t\boldsymbol{X}^+_\Delta{}^t\boldsymbol{X}_\Delta\big(\boldsymbol{y} - {}^t\boldsymbol{X}^+_\Delta\lambda\boldsymbol{\gamma}_\Delta\big) \qquad (2.13) \\
\Leftrightarrow \quad & \hat{\boldsymbol{\beta}}^L_\Delta(\lambda) = \boldsymbol{X}^+_\Delta\boldsymbol{X}_\Delta\boldsymbol{X}^+_\Delta\big(\boldsymbol{y} - {}^t\boldsymbol{X}^+_\Delta\lambda\boldsymbol{\gamma}_\Delta\big) + \boldsymbol{h} \qquad (2.14) \\
\Leftrightarrow \quad & \hat{\boldsymbol{\beta}}^L_\Delta(\lambda) = \boldsymbol{X}^+_\Delta\big(\boldsymbol{y} - {}^t\boldsymbol{X}^+_\Delta\lambda\boldsymbol{\gamma}_\Delta\big) + \boldsymbol{h} \\
\Leftrightarrow \quad & \hat{\boldsymbol{\beta}}^L_\Delta(\lambda) = \underbrace{\boldsymbol{X}^+_\Delta\boldsymbol{y}}_{\boldsymbol{R_1}} - \lambda \times \underbrace{\boldsymbol{X}^+_\Delta{}^t\boldsymbol{X}^+_\Delta\boldsymbol{\gamma}_\Delta}_{\boldsymbol{R_2}} + \boldsymbol{h},
\end{aligned}
$$

avec $\boldsymbol{h}$ un élément du noyau de $\boldsymbol{X}_\Delta$. Si ce noyau se réduit à zéro, alors la solution au problème Lasso est unique. Ce problème est traité par Tibshirani (2012), où l'auteur montre que si les variables sont issues d'une distribution continue, alors ce noyau est toujours réduit au vecteur nul. Nous supposons donc, à partir de maintenant $\boldsymbol{h} = \boldsymbol{0}$ (et donc l'unicité de la solution Lasso).

L'étonnant est ici que la solution $\hat{\boldsymbol{\beta}}^L_\Delta(\lambda)$ est linéaire en λ, pour peu que nous connaissons le support Δ. Une fois cette remarque faite, l'algorithme est naturel. L'idée va être de commencer par $\lambda > 1$ de sorte à avoir $\hat{\boldsymbol{\beta}}^L_\Delta(\lambda) = 0$ et de le faire décroître. En prenant en compte les conditions d'optimalité (équations (2.8)), il est alors possible de chercher la première variable à entrer dans l'ensemble Δ. Une fois celle-ci trouvée, la trajectoire des cœfficients, bien que toujours linéaire, est modifiée. Calculs faits de ces modifications, il est alors possible de chercher la première variable à entrer ou sortir de l'ensemble Δ, lorsque λ continue à décroître. Ainsi, à la fin de ce processus de calcul, nous aurons identifié tout le chemin des solutions (en fonction de λ). Ce chemin est linéaire par morceaux, et les nœuds (*i.e.* les points où la pente change) correspondent chacun à une entrée ou à une sortie d'une variable dans l'ensemble Δ.

L'algorithme est décrit en pseudo-code ci-dessous :

1. Commencer avec $\lambda = +\infty$. Dans ce cas : $\hat{\boldsymbol{\beta}}^L = \boldsymbol{0}_M$.
2. Nous faisons décroître λ vers 0 jusqu'à ce qu'une variable $\boldsymbol{x}_{i_1}$ soit telle que i_1 rejoint Δ.
3. Nous continuons à faire décroître λ vers 0. L'ensemble Δ peut alors changer pour deux raisons :
   — Un indice i_k qui n'appartenait pas à Δ rejoint cet ensemble.
   — Un indice $i_{k'}$ qui appartenait à Δ n'y appartient plus.
4. L'algorithme s'arrête quand $\lambda = 0$. Le problème Lasso pour tout λ est alors résolu.

Il reste maintenant à déterminer quand une variable entre ou sort de cet ensemble.

Entrée et sortie des variables dans Δ

Entrée : L'algorithme LARS-Lasso repose donc sur un ensemble de nœuds, chacun d'entre eux représentant le moment où une variable intègre ou sort de l'ensemble Δ. Supposons que nous nous retrouvons au moment où $\lambda = \lambda(t_k)$.

Une variable $\boldsymbol{x}_i$ qui appartient à l'ensemble Δ sort de celui-ci lorsque (voir équation (2.13)) :

$$\hat{\beta}_i^L(\lambda) = R_{1,i} - \lambda R_{2,i} = 0,$$

ou encore, en supposant $R_{2,i} \neq 0$:

$$\lambda = \frac{R_{1,i}}{R_{2,i}}. \tag{2.15}$$

Sortie : Notons $t_{k,i}^{out}$ le moment de sortie de chaque variable. Le premier moment de sortie est alors calculé de la façon suivante :

$$t_k^{out} = \max_{i \in \Delta} \left\{ t_{k,i}^{out} \,\middle|\, t_{k,i}^{out} < \lambda(t_k) \right\}. \tag{2.16}$$

Une variable $\boldsymbol{X}_i \notin \Delta$ rejoindra l'ensemble Δ dès lors que (voir équation (2.8)) :

$${}^t\boldsymbol{X}_i(\boldsymbol{y} - \boldsymbol{X}\hat{\beta}_i^L(\lambda)) = \pm\lambda. \tag{2.17}$$

En utilisant l'équation (2.13), nous obtenons :

$${}^t\boldsymbol{X}_i\big(\boldsymbol{y} - \boldsymbol{X}(R_{1,i} - \lambda R_{2,i})\big) = \pm\lambda. \tag{2.18}$$

Le moment d'entrée dans l'ensemble Δ est alors :

$$t_{k,i}^{join} = \frac{{}^t\boldsymbol{X}_i\big(\boldsymbol{X}_i\boldsymbol{R}_1 - \boldsymbol{y}\big)}{{}^t\boldsymbol{X}_i\boldsymbol{X}_i\boldsymbol{R}_2 \pm 1}. \tag{2.19}$$

Remarque 25 *$t_{k,i}^{join}$ semble ne pas être bien défini, mais nous choisissons celui des deux qui est dans l'intervalle d'intérêt $[0; \lambda(t_k)]$.*

Conclusion : Soit donc $\lambda(t_k)$ le dernier nœud sur lequel nous nous sommes arrêtés. Le prochain nœud est alors défini, en utilisant (2.16) et (25) par :

$$\lambda(t_{k+1}) = \max(t_k^{join}, t_k^{out}).$$

Nous donnons un exemple de fonctionnement du LARS-Lasso dans la figure 2.5. L'algorithme va d'abord chercher la variable avec laquelle il forme le plus angle. Cette variable étant de plus en plus prise en compte alors que λ diminue, une deuxième variable se rajoute au modèle...

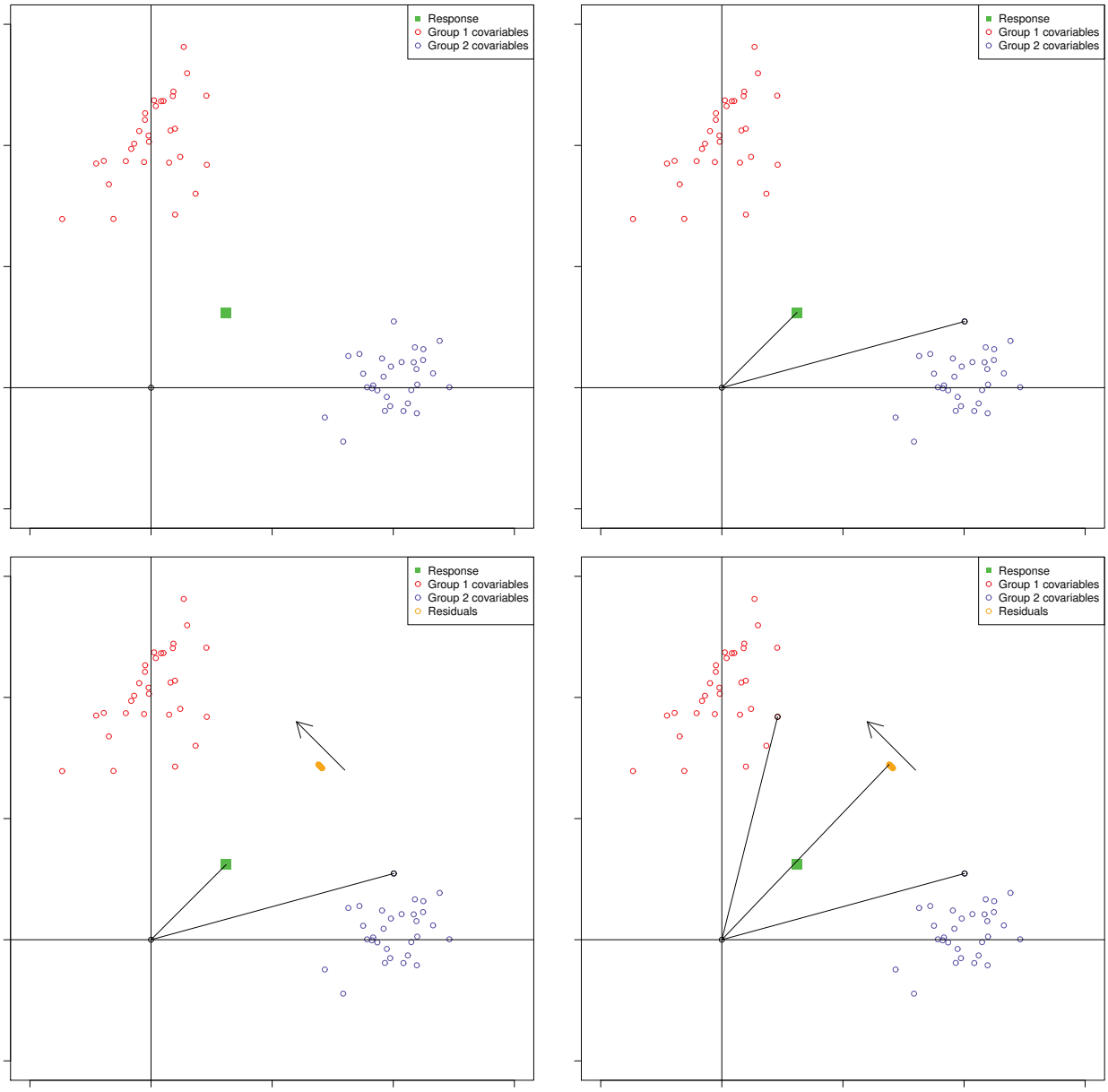


FIGURE 2.5 – *Exemple de fonctionnement de l'algorithme LARS-Lasso en deux dimensions.*

2.6 Résultats théoriques pour le Lasso

Une large littérature est consacrée à l'étude des propriétés théoriques du Lasso. Elle vise à donner des conditions sous lesquelles l'estimateur Lasso permet de retrouver le bon support avec les bons signes et avec une vitesse de convergence suffisante. Nous rappelons dans cette partie les principaux résultats.

2.6.1 Un peu de formalisme

Nous garderons ici les notations déjà établies que nous compléterons. Supposons, pour commencer, que nous connaissons le vrai support de la régression ; autrement dit, supposons connus les prédicteurs dont le cœfficient de régression est non-nul. Notons-le $\mathfrak{S}^*$. Nous décidons de renuméroter les variables du modèle, de sorte que les p premières appartiennent à $\mathfrak{S}^*$ et les $q = M - p$ suivantes n'y appartiennent pas ; ainsi :

$$\boldsymbol{\beta^*} = {}^t(\underbrace{\beta_1^*, ..., \beta_p^*}_{p}, \underbrace{0, .., 0}_{q}) = {}^t(\boldsymbol{\beta}_1^*, \boldsymbol{\beta}_2^*)$$

Les résultats présentés dans cette section seront valables dans le cadre du modèle linéaire, que nous écrivons de la manière suivante :

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.20}$$

où $\boldsymbol{y}$ est le vecteur réponse, $\boldsymbol{X}$ est la matrice des covariables, et $\boldsymbol{\epsilon}$ est une erreur centrée et de variance σ^2.

Nous pouvons d'ores et déjà faire la remarque suivante :

Remarque 26 *Le Lasso donne au plus* $\min(N, M) - 1$ *variables non-nulles.*

Pour voir ceci, il est possible de regarder comment fonctionne l'algorithme LARS, lequel ajoute une à une les variables. La nouvelle variable est choisie en fonction de sa corrélation avec les résidus. Mais, supposons que l'on vient d'ajouter la Nième variable dans le modèle. Le modèle possède alors N vecteurs libres de dimension N, ce qui constitue une base de $\boldsymbol{R}^N$. Ce modèle permet alors d'expliquer toutes les variations de la variable réponse, ce qui laisse les résidus nuls. La nullité des résidus empêche alors de poursuivre l'algorithme.

Notons maintenant respectivement $\boldsymbol{X}_{\mathfrak{S}}$ et $\boldsymbol{X}_{\setminus\mathfrak{S}}$ les matrices constituées des p premières et q dernières colonnes de $\boldsymbol{X}$. Définissons encore la matrice suivante :

$$\boldsymbol{C}^N = {}^t\boldsymbol{X}\boldsymbol{X} = \begin{pmatrix} {}^t\boldsymbol{X}_{\mathfrak{S}}\boldsymbol{X}_{\mathfrak{S}} & {}^t\boldsymbol{X}_{\mathfrak{S}}\boldsymbol{X}_{\setminus\mathfrak{S}} \\ {}^t\boldsymbol{X}_{\setminus\mathfrak{S}}\boldsymbol{X}_{\mathfrak{S}} & {}^t\boldsymbol{X}_{\setminus\mathfrak{S}}\boldsymbol{X}_{\setminus\mathfrak{S}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{C}_{11}^N & \boldsymbol{C}_{12}^N \\ \boldsymbol{C}_{21}^N & \boldsymbol{C}_{22}^N \end{pmatrix}. \tag{2.21}$$

La matrice $\boldsymbol{C}^N$ définit ci-dessus jouera un rôle prépondérant dans les performances du Lasso. Elle permet, entre autre, de définir la matrice de Gram :

Définition 27 (Matrice de Gram) *La matrice de Gram $\mathbf{\Psi}^N$ est définie par :*

$$\mathbf{\Psi}^N = \frac{\boldsymbol{C}^N}{N}.$$

Si $\mathcal{A}_1, \mathcal{A}_2$ sont deux ensembles d'indice, alors nous définissons :

$$\mathbf{\Psi}^N_{\mathcal{A}_1,\mathcal{A}_2} = \frac{{}^t\boldsymbol{X}_{\mathcal{A}_2}\boldsymbol{X}_{\mathcal{A}_1}}{N}.$$

D'ailleurs, comment évaluer les performances d'une régression Lasso ? Nous pouvons discerner trois buts distincts :

— fournir la meilleur approximation du vecteur $\boldsymbol{X}\boldsymbol{\beta}^*$: nous parlerons alors de but de **prédiction**,
— donner la meilleure estimation possible du vecteur $\boldsymbol{\beta}^*$: nous parlerons alors d'objectif d'**estimation**,
— identifier le support $\mathfrak{S}$ de $\boldsymbol{\beta}^*$ (ou encore identifier le vecteur des signes de $\boldsymbol{\beta}^*$) : nous parlerons alors d'objectif de **sélection**.

Dans un modèle de régression classique, une des propriétés classiques recherchée est la consistance. La question est alors de savoir si $\hat{\boldsymbol{\beta}}$ converge vers $\boldsymbol{\beta}^*$ (en loi, en probabilité ou presque sûrement) lorsque la taille de l'échantillon tend vers l'infini. Nous retiendrons la définition suivante :

Définition 28 (Consistance du modèle) *La consistance du modèle est caractérisée par la convergence suivante :*

$$\hat{\boldsymbol{\beta}}^N - \boldsymbol{\beta}^* \underset{N\to\infty}{\to} 0. \tag{2.22}$$

Dans notre cas, cette notion de consistance classique ne semble pas suffisante, puisqu'avec le Lasso nous prétendons pouvoir sélectionner les variables pertinentes. Notons $\hat{\mathfrak{S}}$ l'ensemble des cœfficients non-nuls retenus après inférence. Tout naturellement, cela nous conduit à proposer une consistance en sélection que nous pouvons définir de la manière suivante :

Définition 29 (Consistance en sélection) *La consistance en sélection est caractérisée par la convergence suivante :*

$$\mathbf{P}\left(\hat{\mathfrak{S}}^N = \mathfrak{S}^*\right) \underset{N\to\infty}{\to} 1. \tag{2.23}$$

Enfin, l'utilisateur peut attacher une relative importance au signe des cœfficients de régression, et c'est pourquoi nous définissons :

Définition 30 (Consistance en signe) *La consistance en signe est caractérisée par la convergence suivante :*

$$\mathbf{P}\left(\hat{\boldsymbol{\beta}}^N =_s \boldsymbol{\beta}^*\right) \underset{N\to\infty}{\to} 1, \tag{2.24}$$

où nous avons noté $=_s$ pour signifier que, composante par composante, $sgn(\hat{\boldsymbol{\beta}}^N) = sgn(\boldsymbol{\beta}^*)$.

2.6.2 Cas où N>M

Dans cette partie, le nombre d'observations supérieur au nombre de variables.

La première démonstration de la consistance du modèle pour le Lasso a été faite par Knight et Fu (2000). Avant de poursuivre, il est nécessaire de poser deux hypothèses :

$$\boldsymbol{C}^N \underset{N\to\infty}{\to} \boldsymbol{C} \tag{$\mathfrak{H}_1$}$$

$$\frac{1}{N}\max_{i=1,\dots,N} {}^t\boldsymbol{x}_{i.}\boldsymbol{x}_{i.} \underset{N\to\infty}{\to} 0. \tag{$\mathfrak{H}_2$}$$

L'hypothèse $\mathfrak{H}_2$ conduit généralement à réduire les variables de sorte que les éléments diagonaux de la matrice $\boldsymbol{C}^N$ soient égaux à 1. Le théorème suivant (Knight et Fu 2000) donne la consistance du modèle pour une suite λ_n définie :

Théorème 31 *Supposons que la matrice $\boldsymbol{C}$, telle que définie par $\mathfrak{H}_1$, est non singulière, et supposons que $\lambda_N/N \to \lambda_0 \geqslant 0$, alors*

$$\hat{\boldsymbol{\beta}}^{\boldsymbol{N}} \to_p argmin(\boldsymbol{Z}) \tag{2.25}$$

avec :

$$\boldsymbol{Z}(\boldsymbol{\Phi}) = {}^t(\boldsymbol{\Phi}-\boldsymbol{\beta}^*)\boldsymbol{C}(\boldsymbol{\Phi}-\boldsymbol{\beta}^*) + \lambda_0\sum_{j=1}^{M}|\Phi_j|. \tag{2.26}$$

Remarque 32 *Si $\lambda_n = o(n)$, alors $\hat{\boldsymbol{\beta}}^{\boldsymbol{N}} \to_p argmin({}^t(\boldsymbol{\Phi}-\boldsymbol{\beta}^*)\boldsymbol{C}(\boldsymbol{\Phi}-\boldsymbol{\beta}^*)) = \boldsymbol{\beta}^*$, ce qui montre la consistance du Lasso.*

Remarque 33 *Si $\lambda_0 \neq 0$ alors l'estimateur Lasso comporte un biais.*

Remarque 34 *Si $\lambda_0 = 0$ alors l'estimateur Lasso est consistant en estimation et pour le support.*

La vitesse de convergence de λ_n est donc déterminante dans la consistance de l'estimateur Lasso. Pour peu que la vitesse de convergence soit bien choisie, il est possible d'améliorer le résultat du précédent théorème, et obtenir une $\sqrt{n}$-consistance. La vitesse, serait-elle trop grande, que l'estimateur Lasso ne serait plus consistant ; mais serait-elle trop petite, que l'estimateur Lasso convergerait comme l'estimateur des moindres carrés. Nous avons, plus précisément (Knight et Fu 2000) :

Théorème 35 *Supposons que la matrice $\boldsymbol{C}$, telle que définie par $\mathfrak{H}_1$, est non singulière, et supposons que $\lambda_N/\sqrt{N} \to \lambda_0 \geqslant 0$, alors*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}^{\boldsymbol{N}} - \boldsymbol{\beta}^*) \to_d argmin(\boldsymbol{V}) \tag{2.27}$$

avec :

$$\boldsymbol{V}(\boldsymbol{\Phi}) = -2{}^t\boldsymbol{\Phi}\boldsymbol{W} + {}^t\boldsymbol{\Phi}\boldsymbol{C}\boldsymbol{\Phi} + \lambda_0 \sum_{j=1}^{M} |\Phi_j| I(\beta_j^* = 0) + \Phi_j sgn(\beta_j^*) I(\beta_j^* \neq 0) \tag{2.28}$$

avec :

$$\boldsymbol{W} \sim N(0, \sigma^2 \boldsymbol{C}).$$

Remarquons que lorsque $\lambda_0 = 0$, l'estimateur Lasso converge vers l'estimateur des moindres carrés ordinaires, et le terme de pénalisation n'intervient plus :

$$\text{argmin}(\boldsymbol{V}) = \boldsymbol{C}^{-1}\boldsymbol{W} \sim N(0, \sigma^2 \boldsymbol{C}^{-1}).$$

Si nous avons une vitesse optimale en terme de convergence en estimation, l'estimateur n'est donc plus consistant en terme de sélection de variable.

2.6.3 Cas $M \geqslant N$

Dans le cas $M < N$, quelques conditions suffisent pour que l'estimateur Lasso soit consistant, tant en signe qu'en sélection. Le cas $M \geqslant N$, rencontré bien plus souvent en pratique, est plus problématique. Il va nécessiter souvent deux types d'hypothèses, la première sur la matrice de Gram (induisant ainsi une corrélation maîtrisée) et la seconde sur la puissance minimale du signal, caractérisée par le plus petit cœfficient non-nul de $\boldsymbol{\beta}^*$.

Pour cette section, les résultats les plus intéressants ont été publiés par Zhao et Yu (2006). Dans toute cette section, nous nous plaçons sous les hypothèses classiques de régularité $\mathfrak{H}_1$ et $\mathfrak{H}_2$.

Il est nécessaire d'introduire de nouvelles définitions, que nous donnons ici dans un premier temps, avant de nous permettre quelques mots d'explications.

Définition 36 (Fortement consistant en signe (CS+)) *Le Lasso sera dit fortement consistant en signe s'il existe une suite λ_n, qui étant une fonction de n indépendante de $\boldsymbol{y_n}$ et $\boldsymbol{X_n}$ est telle que :*

$$\lim_{N\to\infty} P(\hat{\boldsymbol{\beta}}^N(\lambda_N) =_s \boldsymbol{\beta}^*) = 1.$$

Définition 37 (Généralement consistant en signe (CS-)) *Le Lasso sera dit généralement consistant en signe si :*

$$\lim_{N\to\infty} P(\exists \lambda \geqslant 0,\ \hat{\boldsymbol{\beta}}^N(\lambda) =_s \boldsymbol{\beta}_N) = 1.$$

En reprenant les notations introduites dans l'équation (2.21), nous définissons encore :

Définition 38 (Condition d'irreprésentabilité forte (CI+)) *Il existe un vecteur constant positif $\boldsymbol{\eta}$ tel que :*

$$|\boldsymbol{C}_{21}^N(\boldsymbol{C}_{11}^N)^{-1}sgn(\boldsymbol{\beta}_1^*)| \leqslant \mathbf{1} - \boldsymbol{\eta},$$

où l'inégalité doit être comprise terme à terme.

Définition 39 (Condition d'irreprésentabilité faible (CI-)) *Il existe un vecteur constant positif* $\boldsymbol{\eta}$ *tel que :*

$$|\boldsymbol{C}_{21}^N(\boldsymbol{C}_{11}^N)^{-1}sgn(\boldsymbol{\beta}_1^*)| < 1,$$

où l'inégalité doit être comprise terme à terme.

La dénomination de ces différentes propriétés est logique dans le sens où :

$$CS+ \Rightarrow CS- \quad \text{et} \quad CI+ \Rightarrow CI-.$$

En effet, la condition CS+ implique qu'une séquence prédéterminée de régularisation peut être choisie *a priori* afin d'obtenir la consistance en signe, alors que la condition CS- implique seulement l'existence d'une telle séquence. La deuxième implication est évidente.

La condition CI peut être vue d'une manière plus éclairante :

$$\begin{aligned} {}^t(\boldsymbol{C}_{21}^N(\boldsymbol{C}_{11}^N)^{-1}) &= {}^t({}^t\boldsymbol{X}_{\backslash\mathfrak{S}}\boldsymbol{X}_{\mathfrak{S}}({}^t\boldsymbol{X}_{\mathfrak{S}}\boldsymbol{X}_{\mathfrak{S}})^{-1}) \\ &= ({}^t\boldsymbol{X}_{\mathfrak{S}}\boldsymbol{X}_{\mathfrak{S}})^{-1} \times {}^t\boldsymbol{X}_{\mathfrak{S}}\boldsymbol{X}_{\backslash\mathfrak{S}}. \end{aligned}$$

Nous reconnaissons ici la formule classique de l'estimateur des moindres carrés ordinaires. Il s'agit donc de régresser chaque vecteur n'appartenant pas au support réel de la régression sur les vecteurs y appartenant. Supposons que $sgn(\boldsymbol{\beta}_1^*)$ soit de signe positif pour chaque composante, alors les CI stipulent simplement que la valeur absolue de la somme des cœfficients des régressions des vecteurs qui ne sont pas dans le support $\mathfrak{S}^*$ sur l'ensemble des vecteurs qui y sont doit être plus petite que 1.

Nous avons alors le théorème suivant, démontré par Zhao et Yu (2006) :

Théorème 40 *Si p et q sont constants, et sous certaines conditions de régularité (voir Zhao et Yu (2006)), nous avons :*

$$CI+ \Rightarrow CS+ \Rightarrow CS- \Rightarrow CI-.$$

Cette suite d'implications doit être vue comme un théorème (première implication) suivi d'une réciproque affaiblie (troisième implication). Notons que des résultats similaires ont été obtenus indépendamment par Meinshausen et Bühlmann (2006).

Ce théorème n'a rien d'étonnant, et semble même plutôt intuitif. Regardons sur un exemple simple à quoi aboutit cette condition.

Exemple 41 *Posons :*

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0 & \cos(\theta) \\ 0 & 1 & \sin(\theta) \end{pmatrix}$$

et :

$$\boldsymbol{Y} = \frac{\boldsymbol{x}_{.1} + \boldsymbol{x}_{.2}}{\|\boldsymbol{x}_{.1} + \boldsymbol{x}_{.2}\|_2}.$$

Notons que tous les vecteurs de cet exemple sont normés à 1, et que le troisième vecteur, dépendant de θ décrit toute la boule unité. Voyons pour quelles valeurs de θ la condition d'irreprésentabilité est respectée. Puisque $sgn(\boldsymbol{\beta})$ est toujours positif, regardons la valeur de la valeur absolue de la régression de $\boldsymbol{x}_{.3}$ sur $\boldsymbol{x}_{.1}$ et $\boldsymbol{x}_{.2}$.

Un rapide calcul donne pour cœfficients de cette dernière régression :

$$\begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}.$$

Il suffit de regarder quand $|\cos(\theta) + \sin(\theta)| < 1$. Il est bien connu que cela est vrai lorsque $\theta \in]\pi k - \frac{\pi}{2}; \pi k[$ avec $k \in \mathbb{N}$.

Au prix de deux hypothèses supplémentaires (en sus de la condition CI+) nous pouvons encore obtenir un meilleur résultat. La première porte sur la valeur propre minimale de la matrice de Gram restreinte :

$$\exists C > 0 \quad \text{t.q.} \quad vp_{min}(\boldsymbol{\Psi}_{\mathfrak{S}^*,\mathfrak{S}^*}) \geqslant C, \tag{2.29}$$

avec vp_{min} désignant la valeur propre minimum de la matrice. La deuxième hypothèse porte sur la valeur β_{min} du plus petit cœfficient non-nul de $\boldsymbol{\beta}^*$:

$$\beta_{min} > \lambda_N \left(\|\boldsymbol{\Psi}^{-1}_{\mathfrak{S}^*,\mathfrak{S}^*}\|_\infty + \frac{4\sigma}{\sqrt{vp_{min}(\boldsymbol{\Psi}_{\mathfrak{S}^*,\mathfrak{S}^*})}} \right). \tag{2.30}$$

Nous pouvons maintenant énoncer le théorème, démontré dans (Wainwright 2009) :

Théorème 42 *Supposons que la condition d'irreprésentabilité est vérifiée, ainsi que les conditions données par les équations 2.29 et 2.30. Supposons également que la suite de paramètres de régularisation satisfait :*

$$\lambda_N > \frac{2}{\eta}\sqrt{\frac{\sigma^2 \log M}{N}}.$$

La régression Lasso parvient alors à déterminer le bon support, avec les bons signes, avec une probabilité qui tend vers 1 quand N tend vers l'infini.

2.6.4 Limites théoriques de la sélection de variables

Dans cette partie, nous changeons légèrement le cadre d'étude. Nous conservons le modèle linéaire tel que défini par l'équation 2.20 mais nous allons supposer que les variables colonnes $\boldsymbol{x}_i$ de la matrice $\boldsymbol{X}$ sont i.i.d. et sont telles que :

$$x_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}),$$

pour une matrice $\mathbf{\Sigma}$ donnée.

Tout d'abord, définissons, comme dans Wainwright (2009), un estimateur de recherche exhaustive $\hat{\beta}^{exh}$. Il est donné par l'algorithme suivant :

1. Pour chaque sous-ensemble T dans l'ensemble des parties à p éléments dans $\{1, ..., M\}$ (c'est-à-dire tous les ensembles ayant un cardinal égal au nombre d'éléments non-nuls dans β^*), nous définissons :

$$Z(T) = \min_{\beta_T \in \mathbb{R}^p} \|y - X_T \beta_T\|_2^2$$

2. L'estimateur $\hat{\beta}^{exh}$ est alors défini par le sous-ensemble T qui minimise $Z(T)$, avec T un ensemble dans l'ensemble des parties à p éléments dans $\{1, ..., M\}$.

Il apparaît alors que cet estimateur exhaustif peut, sous certaines conditions sur la puissance minimale du signal, sur σ^2, sur la matrice $\mathbf{\Sigma}$, et sur N, M, p retrouver le support recherché. Une condition essentielle est que $p \log M$ soit petit par rapport à N. Dans le cas inverse, nous aboutissons à ce que Verzelen *et al.* (2012) appellent l'ultra-haute dimension. Et dans ce cadre là, ils ont prouvé, qu'à moins d'une condition d'une valeur minimale du plus petit cœfficient non-nul de β^* qui explose, il est impossible de retrouver le bon support. Pire encore, il a démontré qu'il n'est même pas possible de réduire la dimensionnalité du problème, c'est-à-dire un ensemble qui contiendrait le support recherché. L'ultra-haute dimension apparaît donc comme un mur - pour l'instant - infranchissable.

2.7 Alternatives et améliorations du Lasso

Trois problèmes majeurs sont soulevés par l'utilisation du Lasso, et chaque alternative permet de répondre, au moins partiellement, à un de ces problèmes :

1. Le biais,
2. L'inconsistance de la sélection en présence de corrélation,
3. Le manque de test sur les cœfficients obtenus.

Le biais est un problème dont nous avons déjà vus les effets. En effet, dans le cadre d'un design orthogonal, nous avions vu que les cœfficients Lasso étaient les cœfficients des moindres carrés "contraints" vers 0 par un réel constant. Que le prédicteur appartienne ou non au support de la régression, cela ne change rien : il est translaté vers 0.

Pour comprendre le lien entre inconsistance en sélection et présence de corrélation, il suffit de regarder les conditions CI.

Enfin, le manque de test est lié au fait que nous ne connaissons pas la distribution des estimateurs, pour laquelle il faudrait au moins connaître *a priori* le support de régression.

2.7.1 Réduire le biais

Pour réduire le biais du Lasso, une solution paraît simple et intuitive. Une régression Lasso est faite dans un premier pour choisir le support de la régression avant de refaire une régression classique sur les prédicteurs sélectionnés. Le biais étant ainsi réduit, libre à l'utilisateur de poursuivre son analyse par des tests classiques en régression. De cette manière, en supposant le support de la prédiction correctement choisi, nous obtenons toutes les propriétés d'une régression classique.

Cette idée a été développée par Meinshausen (2007). Il propose une procédure en deux étapes, dont la première est une régression classique simple, et dont la deuxième consiste à refaire une régression Lasso sur l'ensemble des prédicteurs sélectionnés par la première régression Lasso.

Une méthode unifiée, proposée par Zou (2006), est l'adaptative Lasso. Cette méthode est une régression Lasso pondérée par l'inverse des cœfficients obtenus par régression simple de chacun des prédicteurs sur la variable réponse.

2.7.2 Lasso et corrélation

Le problème de la corrélation est sans doute un des plus problématiques. En tout état de cause, plus le nombre de variables grandit rapidement par rapport au nombre d'observations, plus la corrélation entre les données sera élevée.

Les performances du Lasso sont liées à la cohérence mutuelle du modèle, qui peut être définie comme suit (Donoho *et al.* 2006) :

Définition 43 *Appelons $\boldsymbol{G} = {}^t\boldsymbol{X}\boldsymbol{X}$. En supposant normé par la norme deux les prédicteurs (et donc, $\boldsymbol{G}$ a des 1 sur la diagonale) la cohérence mutuelle est alors définie par :*

$$M(\boldsymbol{X}) = \max_{i \neq j} |\boldsymbol{G}_{ij}|$$

Il est alors possible de montrer que le bon support, qui est de cardinal p peut être retrouvé si :

$$p \leqslant \frac{\frac{1}{M(\boldsymbol{X})} + 1}{4}.$$

Cette condition est plutôt restrictive, comme le montre l'exemple suivant :

Exemple 44 *Prenons le cas où J variables de dimensions 25 forment la matrice $\boldsymbol{X}$, et supposons que chacun de ces vecteurs est composé d'une répétition indépen-*

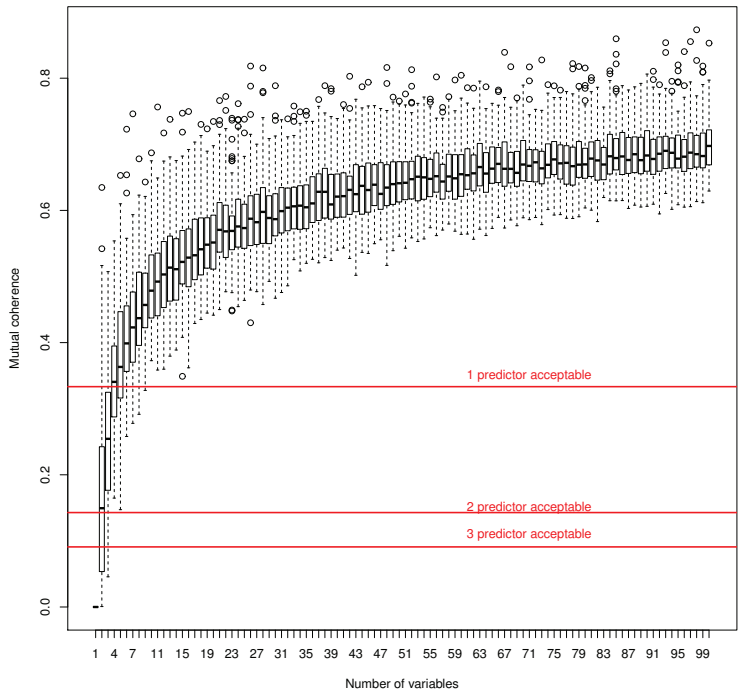


FIGURE 2.6 – *Évolution de la cohérence mutuelle en fonction du nombre de variables, pour un nombre d'observations fixé à 25.*

dante de loi uniforme entre -1 et 1. La figure 2.6 suivant montre l'évolution de la cohérence mutuelle dans un tel exemple. Au-delà de cinq prédicteurs dans le modèle, p doit être égal à zéro (i.e. : aucun prédicteur n'est censé appartenir au support réel de la régression).

Il existe trois solutions classiques pour répondre au problème de la corrélation dans le cas du Lasso, selon le type de problème :

— Les variables sont ordonnées : *fuzed* Lasso (Tibshirani *et al.* 2005). Le modèle est alors le suivant :

$$\hat{\boldsymbol{\beta}}^{fLasso}(\lambda_1,\lambda_2) = \underset{\boldsymbol{\beta}\in\mathbb{R}^M}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left(y_n - \sum_{m=1}^{M} \beta_m x_{nm} \right)^2 + \lambda_1 \sum_{m=1}^{M} |\beta_m| + \lambda_2 \sum_{m=2}^{M} |\beta_m - \beta_{m-1}| \right]. \quad (2.31)$$

Les variables étant ordonnées et étant attendu que les cœfficients des régresseurs ne varient pas "trop" entre une covariable et la covariable précédente, la pénalisation de leur différence apparaît adaptée.

— Les groupes de variables corrélées sont connus : *grouped* Lasso (Yuan et Lin 2006). Supposons que nous avons séparé nos prédicteurs en L groupes. Alors le modèle est le suivant :

$$\hat{\boldsymbol{\beta}}^{grLasso}(\lambda_1) = \underset{\boldsymbol{\beta}\in\mathbb{R}^M}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left(y_n - \sum_{m=1}^{M} \beta_m x_{nm} \right)^2 + \lambda_1 \sum_{l=1}^{L} p_l \|\boldsymbol{\beta}_l\|_2 \right], \quad (2.32)$$

où p_l permet de pondérer en fonction de la taille de chacun des groupes, et $\boldsymbol{\beta}_l$ représente le vecteur de l'ensemble des cœfficients du groupe l. Il faut noter que la norme deux utilisée n'est pas au carré. Il a été montré (Yuan et Lin 2006) que ce modèle permet de faire de la sélection de variables par groupe. Notons qu'il existe également le *sparse group* Lasso dont le modèle s'écrit comme dans (2.32), mais où une pénalité Lasso classique est rajoutée (Friedman *et al.* 2010). Cela permet de faire d'une part de la sélection au niveau des groupes, et ensuite, une sélection des variables à l'intérieur de chaque groupe.

— Les groupes de variables corrélées ne sont pas connus : *elastic* Net (Zou et Hastie 2005). Le modèle est le suivant :

$$\hat{\boldsymbol{\beta}}^{elasnet}(\lambda_1,\lambda_2) = \underset{\boldsymbol{\beta}\in\mathbb{R}^M}{\operatorname{argmin}} \left[\sum_{n=1}^{N} \left(y_n - \sum_{m=1}^{M} \beta_m x_{nm} \right)^2 + \lambda_1 \sum_{m=1}^{M} |\beta_m| + \lambda_2 \|\boldsymbol{\beta}\|_2^2 \right]. \quad (2.33)$$

Il a été montré que cette méthode, qui est un mélange de régression Lasso et de régression Ridge, jouit des avantages des deux méthodes. Par la pénalisation Lasso, une sélection de variable est faite, tandis que par la pénalisation Ridge les covariables corrélées auront tendance à avoir des cœfficients de régression proches.

2.7.3 Tests pour le Lasso

Comme nous l'avons déjà dit, aucun test exact n'a été proposé à ce jour pour étudier les cœfficients obtenus par la méthode du Lasso. Cependant, utilisant les capacités de plus en plus performantes des outils informatiques, deux techniques ont vu le jour :

1. le bootstrap,
2. le resampling.

Il est important de ne pas confondre ces deux idées, dont l'appellation peut cependant prêter à confusions. Le bootstrap consiste en un rééchantillonage au niveau des individus. L'idée est de créer de nouveaux échantillons à partir de l'échantillon initial qui permettront de donner des indications sur les cœfficients estimés au départ. Le resampling part de l'idée qu'il est possible de déduire des propriétés pour le modèle intial.

Le bootstrap est une technique qui n'est pas spécifique au Lasso ; pour plus d'indication voir le livre de Efron (1982). Dans le cadre de la régression Lasso, le bootstrap a d'abord été évoqué par Knight et Fu (2000), mais l'étude la plus complète de la méthode revient à Bach (2008) complété par Bach (2009) créant ce qu'il appelle le boLasso. Deux types de bootstrap y sont proposés. Le premier est un bootstrap dans lequel nous tirons avec remise un échantillon de même taille pour obtenir des couples $\boldsymbol{y}^*$ et $\boldsymbol{X}^*$. Le second, dont les résultats théoriques et applications numériques présentés par Bach (2009) montrent qu'il est plus efficace (sous certaines conditions) consiste d'abord à calculer les résidus :

$$y = \hat{y} + \hat{\varepsilon}.$$

Ce sont alors les résidus $\hat{\varepsilon}$ qui sont rééchantillonnés (tirage avec remise) pour obtenir des $\hat{\varepsilon}^*$. On calcule alors :

$$y^* = \hat{y} + \hat{\varepsilon}^*.$$

2.7.4 Exemple

Pour cet exemple, nous avons choisi de prendre dix variables pour cinq observations. Chacune des neufs premières variables est issue de cinq variables indépendantes de loi uniforme sur $[-1;1]$ tandis que la dixième variable est calculée comme suit :

$$x_{.10} = x_{.1} + \varepsilon,$$

où ε est un vecteur dont chaque composante est issue d'une loi normale de moyenne nulle et d'écart type 0.01. Chaque prédicteur est ensuite centré, puis réduit au sens de la norme 2. La matrice ${}^tXX$ vaut dans notre cas :

$$\begin{pmatrix}
1.00 & 0.13 & 0.25 & 0.97 & 0.52 & 0.81 & 0.68 & 0.45 & 0.76 & 1-10^{-4} \\
0.13 & 1.00 & -0.20 & 0.28 & -0.22 & -0.07 & 0.25 & -0.78 & 0.09 & 0.14 \\
0.25 & -0.20 & 1.00 & 0.05 & -0.49 & 0.76 & -0.27 & 0.44 & -0.39 & 0.25 \\
0.97 & 0.28 & 0.05 & 1.00 & 0.61 & 0.68 & 0.82 & 0.26 & 0.86 & 0.97 \\
0.52 & -0.22 & -0.49 & 0.61 & 1.00 & 0.11 & 0.82 & 0.28 & 0.91 & 0.51 \\
0.81 & -0.07 & 0.76 & 0.68 & 0.11 & 1.00 & 0.35 & 0.57 & 0.30 & 0.81 \\
0.68 & 0.25 & -0.27 & 0.82 & 0.82 & 0.35 & 1.00 & -0.04 & 0.90 & 0.69 \\
0.45 & -0.78 & 0.44 & 0.26 & 0.28 & 0.57 & -0.04 & 1.00 & 0.21 & 0.44 \\
0.76 & 0.09 & -0.39 & 0.86 & 0.91 & 0.30 & 0.90 & 0.21 & 1.00 & 0.76 \\
1-10^{-4} & 0.14 & 0.25 & 0.97 & 0.51 & 0.81 & 0.69 & 0.44 & 0.76 & 1.00
\end{pmatrix}.$$

La cohérence mutuelle de cette matrice vaut $1-10^{-4}$, ce qui ne permet d'aboutir à aucun résultat théorique. Nous allons étudier les deux modèles suivants :

$$y_1 = \frac{x_{.2} + x_{.3}}{2} + \varepsilon_1, \tag{2.34}$$

et :

$$y_2 = \frac{x_{.1} + x_{.3}}{2} + \varepsilon_2, \tag{2.35}$$

où ε_1 et ε_2 représentent un bruit blanc gaussien d'écart type 0.01. Le premier modèle a été choisi pour représenter le "bon" cas, où le problème de la corrélation est limité, tandis que le second a été choisi pour représenter le "mauvais" cas. Plus formellement, si nous regardons le maximum du vecteur issu des CS (+ ou -) nous obtenons 0.88 pour le premier modèle et 1.09 pour le second. Or dans les CI (+ ou -) il est demandé à ce maximum d'être plus petit que un. Le premier modèle respecte donc ces conditions tandis que le deuxième non.

Nous commençons notre étude par effectuer une régression Lasso simple, Figure 2.7. Dans le modèle 1, le bon support est retrouvé tandis que dans

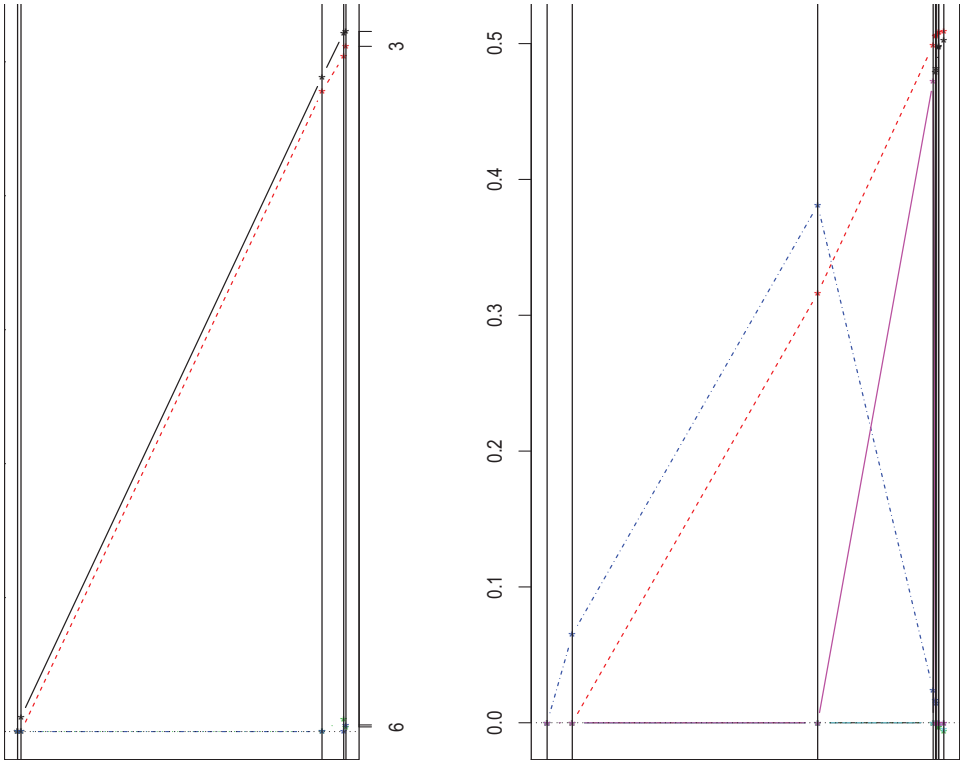


FIGURE 2.7 – *Evolution des cœfficients de régression en fonction du paramètre λ. A gauche le modèle 1 dans lequel les deux covariables du support se détachent nettement (rouge et noir). A droite le modèle deux, dans lequel la deuxième variable d'intérêt $\boldsymbol{x}_{.2}$ voit son cœfficient se détacher (rouge) tandis que la première variable d'intérêt (rose) apparait de façon alternative avec la covariable 10 (en bleue).*

le modèle 2 les covariables 1 et 10 apparaissent alternativement.

Pour le modèle 1, seul reste le problème du biais. Voyons alors comment l'adaptative Lasso permet de régler ce problème. Nous choisissons, *a posteriori*, le meilleur degré de pénalisation à 0.01 près. Nous obtenons alors les cœfficients suivants :

— Lasso : 0.4850551 et 0.4742315.
— Adaptative Lasso : 0.5064834 et 0.4952913.
— Régression simple sur les prédicteurs d'intérêt : 0.5240 et 0.5131.

Cela permet par l'exemple de vérifier les résultats théoriques de l'adaptative Lasso : le biais est bien réduit !

Concentrons-nous maintenant sur le deuxième modèle. L'utilisation du grouped Lasso ou de l'Elastic Net permet, comme montré dans les Figures 2.8 et 2.9, de sélectionner en même temps les prédicteurs corrélés $\boldsymbol{x}_{.1}$ et $\boldsymbol{x}_{.10}$. On note cependant l'apparition du prédicteur $\boldsymbol{x}_{.4}$: est-ce étonnant ? Pour répondre à cette question, regardons, une fois de plus, la matrice ${}^t\boldsymbol{X}\boldsymbol{X}$, et plus précisément l'élément de la quatrième ligne et de la première colonne : 0.97. Cette forte corrélation, dont nous n'avions pas pris note jusqu'à présent, est maintenant révélée. L'intéressant est qu'en utilisant l'elastic net, méthode qui est sans *a priori* sur la structure de corrélation des données,

nous sélectionnons avec un cœfficient qui tend à être égal les prédicteurs 1, 10 et 4, tandis que le grouped Lasso (dans lequel chaque prédicteur forme un groupe, mis à part les prédicteurs 1 et 10 qui sont unis dans un même groupe) aura tendance à sélectionner 1 et 10 avec un cœfficient égal et à délaisser le prédicteur 4.

2.8 Conclusion

Le Chapitre 2 nous a permis de présenter une théorie statistique, celle de la régression pénalisée Lasso, que nous serons amenés à utiliser très largement au cours de cette thèse. Cela nous a permis d'en percevoir les enjeux et les limitations. Il sera alors plus aisé pour le lecteur de comprendre les résultats - et leur portée - qui seront donnés dans les chapitres à venir. Ce Chapitre 2 pourra également être vu comme une motivation au Chapitre 8, dans lequel nous proposons un algorithme permettant de lever certaines difficultés rencontrées avec la régression de type Lasso. En particulier, comme nous venons de le voir, le problème de la corrélation des données, n'est pas entièrement résolu par la littérature.

Avant d'entamer la partie II, qui rassemble la partie la plus importante de nos résultats originaux, nous nous proposons, dans le chapitre suivant, de décrire et de faire une première analyse du jeu de données sur lequel nous avons travaillé tout au long de cette thèse. Nous expliquerons également le modèle biologique duquel il est issu.

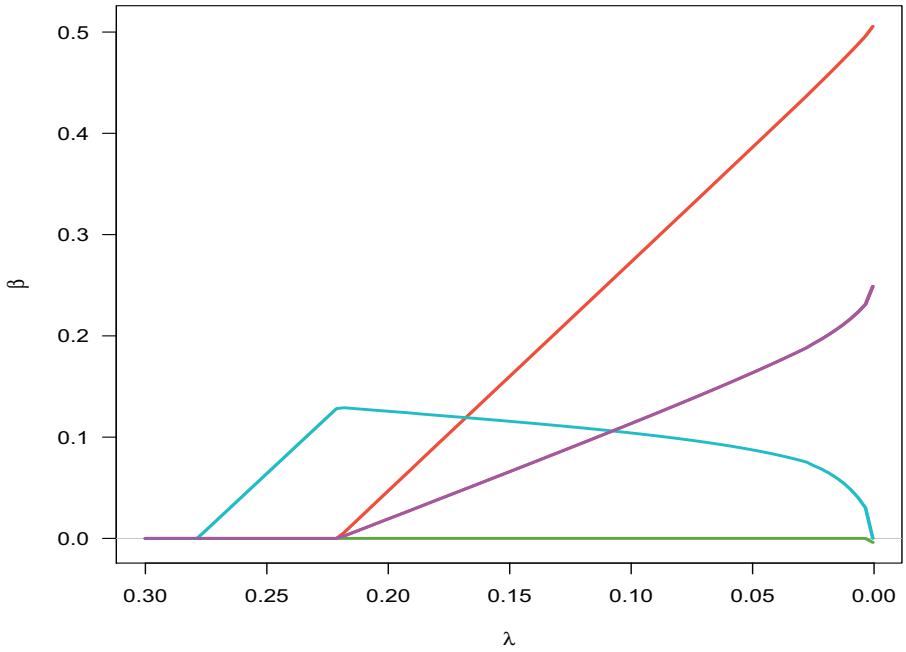


FIGURE 2.8 – *Evolution des cœfficients de régression en fonction du paramètre λ. La courbe violette représente les deux covariables correllées* $\boldsymbol{x}_{.1}$ *et* $\boldsymbol{x}_{.10}$*, la rouge la deuxième variable d'intérêt du modèle* $\boldsymbol{x}_{.2}$ *et en cyan une variable hors d'intérêt* $\boldsymbol{x}_{.4}$*.*

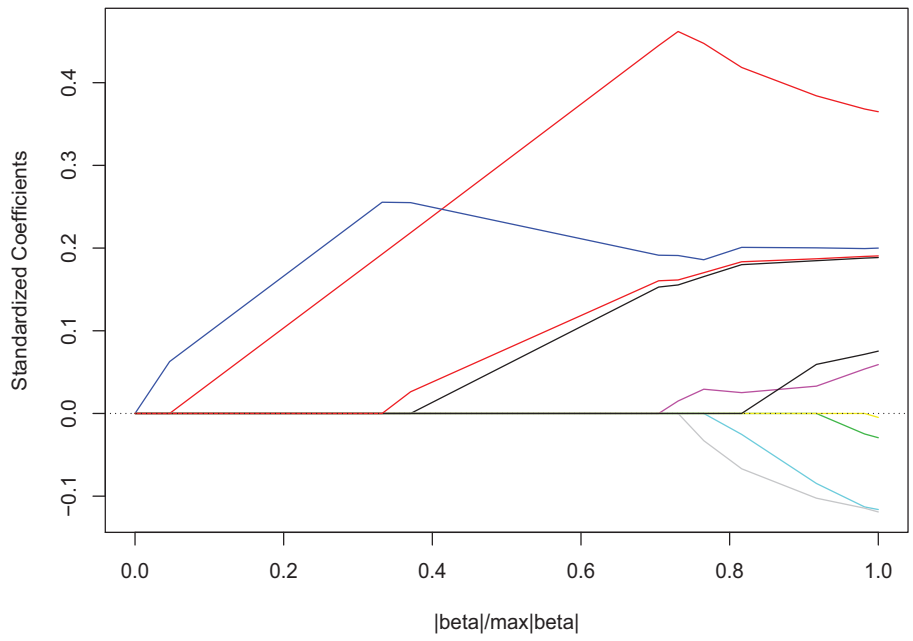


FIGURE 2.9 – *Evolution des cœfficients de régression en fonction du paramètre λ De haut en bas :* $\boldsymbol{x}_{.2}$*, puis, groupés* $\boldsymbol{x}_{.1}$*,* $\boldsymbol{x}_{.4}$ *et* $\boldsymbol{x}_{.10}$ *puis quelques variables hors d'intérêts.*

Modèle biologique

3

"The revolution in cancer research can be summed up in a single sentence : cancer is, in essence, a genetic disease. In the last decade, many important genes responsible for the genesis of various cancers have been discovered, their mutations precisely identified, and the pathways through which they act characterized." (Vogelstein et Kinzler 2004)

Le cancer est une maladie essentiellement génétique. Il y a donc, chez les patients souffrant d'un cancer, une modification de leur programme génique. Comme nous l'avons vu dans le Chapitre 1, ce programme génique est un système complexe. Le cancer s'inscrit donc parfaitement dans le cadre que nous avons défini dans le premier chapitre. C'est une maladie qui est liée à une modification du programme génique, laquelle a des répercussions phénotypiques importantes.

Il est donc intéressant, étant donné deux individus, l'un étant sain et l'autre étant atteint par un cancer, de comprendre quelles modifications du programme génique ont conduit l'individu malade vers son état cancéreux. Nous pourrions ensuite proposer des interventions pour tenter de modifier de façon orientée le programme génique cancéreux pour lui faire regagner son état initial. C'est cette idée qui a motivé tous les travaux présentés dans cette thèse. L'exemple biologique qui a servi de support à ce travail est la leucémie lymphoïde chronique, maladie pour laquelle nous faisons quelques rappels dans la partie suivante.

3.1 La leucémie lymphoïde chronique comme modèle biologique

La leucémie lymphoïde chronique est la leucémie la plus fréquente chez l'adulte dans les pays occidentaux (Ghia *et al.* 2007). L'âge moyen de diagnostic de cette maladie est de 50 ans, et sa prévalence est estimée entre 30 et 50 pour 100 000. La progression de cette maladie est insidieuse et les premières étapes de son développement sont généralement asymptomatiques (Chiorazzi *et al.* 2005).

La leucémie lymphoïde chronique est une maladie touchant des cellules du sang appelées lymphocytes B. Ces cellules, produites dans la mœlle osseuse, ont un rôle important dans le processus de défense immunitaire, lequel assure la défense du corps humain contre les éléments pathogènes exté-

rieurs. Elle se caractérise par une accumulation de ces lymphocytes B dans la mœelle osseuse, le sang, et la lymphe (Rozman 1995, Zenz *et al.* 2009). La particularité de cette maladie est que son évolution clinique présente une forte hétérogénéité (Schroers *et al.* 2005). En effet, si certains patients présentent une forme agressive de la maladie qui nécessite un traitement précoce, la plupart d'entre eux présente une forme indolente, qui, dans certains cas, ne demande même aucun traitement.

De manière plus précise, deux sous-groupes de patients ont été identifiés. En effet, il apparaît que la présence de mutation somatique dans les chaînes d'immunoglobuline influence significativement le pronostic pour le patient. En particulier, les patients dont la partie variable des chaînes lourdes de l'immunoglobuline(IGvH) reste non-mutée ont une forme plus agressive de la maladie (Hamblin *et al.* 1999, Damle *et al.* 1999). Certains gènes permettent de discriminer les patients qui ont une forme mutée ou non de l'IGvH. Le plus remarquable d'entre eux est sans doute le gène ZAP70, dont la protéine associée est mesurée. Cette dernière protéine a une expression plus élevée chez les patients ayant la forme non mutée de l'IGvH (Rassenti *et al.* 2004). Dans la suite, nous aurons donc trois types d'individus :

— les individus sains,
— les individus atteints de la LLC, état indolent,
— les individus atteints de la LLC, état agressif.

Cette maladie se caractérise donc par une prolifération incontrôlée de lymphocytes B aboutissant à un cancer incurable (Chiorazzi *et al.* 2005). Le mécanisme ainsi que les raisons de cette prolifération ne sont pas encore bien compris à l'heure actuelle. Cependant, l'hypothèse d'une stimulation antigénique chronique de certains lymphocytes est la plus probable (Stevenson et Caligaris-Cappio 2004, Chiorazzi *et al.* 2005). Cette stimulation se fait grâce à un récepteur spécifique à la surface des lymphocytes B appelé BCR ("B-Cell Receptor").

3.2 Plan d'expérience de l'étude

Le plan d'expérience et toutes les précisions utiles sur les manipulations ont été décrits dans (Vallat *et al.* 2007). Nous en faisons ici un résumé (voir Figure 3.1).

Douze patients atteints de la leucémie lymphoïde chronique ont été inclus dans l'étude, ainsi que six individus sains. Tous les patients malades ont été testés afin de connaître le statut (muté/non muté) de la partie variable des chaînes lourdes des immunoglobulines ainsi que le niveau d'expression de la protéine ZAP70. Cela a permis de former deux sous groupes parmi les patients atteint de LLC, six ayant la forme indolente de la maladie tandis que les six autres ont la forme agressive.

Des cellules ont ensuite été prélevées. À chaque temps, les cellules ont été divisées en deux groupes. Dans le premier, une stimulation du BCR a

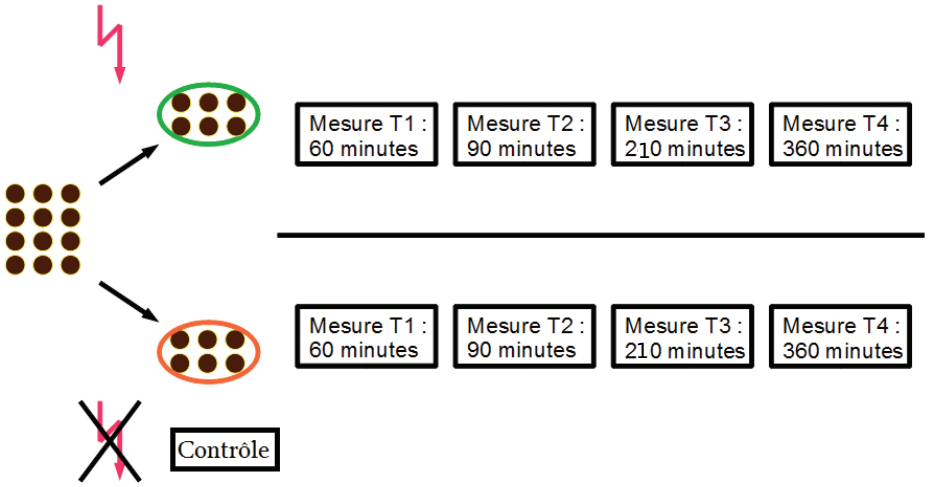


FIGURE 3.1 – *Image venant d'une micropuce à ADN. Chaque point correspond à un probeset particulier.*

été effectuée, tandis que dans le second aucun stimulation n'est faite. Cette stimulation du BCR marque le t_0 de l'étude. A deux temps précoces (60 et 90 minutes) et à deux temps plus tardifs (210 et 390 minutes) des cellules ont été prélevées des deux groupes de cellules et la quantité d'ARNm de chaque gène a été mesurée via des micropuces à ADN. Ce qu'il faut savoir, c'est que les puces à ADN ne mesurent pas directement les gènes, mais des portions de gènes appelés probesets. Certains gènes sont donc représentés par plusieurs probesets, ce qui explique que nous ayons plus de probesets que de gènes : 54 675.

Un dysfonctionnement de matériel a conduit à retirer de l'étude un des patients ayant la version agressive de la maladie.

3.2.1 Puces à ADN

Les puces à ADN, ou microarrays, sont une technique pour mesurer l'expression de multiples gènes au même instant. Une puce à ADN est un ensemble de courts fragments d'ADN - appelés "sondes" - fixées en rangées ordonnées sur une petite surface qui est du verre dans la plupart des cas. Cette biotechnologie récente (utilisée depuis une dizaine d'années) permet d'analyser le niveau d'expression des gènes (transcrits) dans une cellule, un tissu, un organe, un organisme ou encore un mélange complexe, à un moment donné et dans un état donné par rapport à un échantillon de référence.

Le principe de la puce à ADN repose sur la propriété que possède l'ADN dénaturé simple brin de s'apparier avec un brin de séquence complémentaire.

Concrètement chaque gène est découpé en plusieurs probesets, correspondant à des régions spécifiques de celui-ci. Pour chaque probe-set, un probeset de séquence proche est créé en modifiant un des nucléotides de sorte à créer une séquence quasi-similaire mais qui ne correspond à rien de

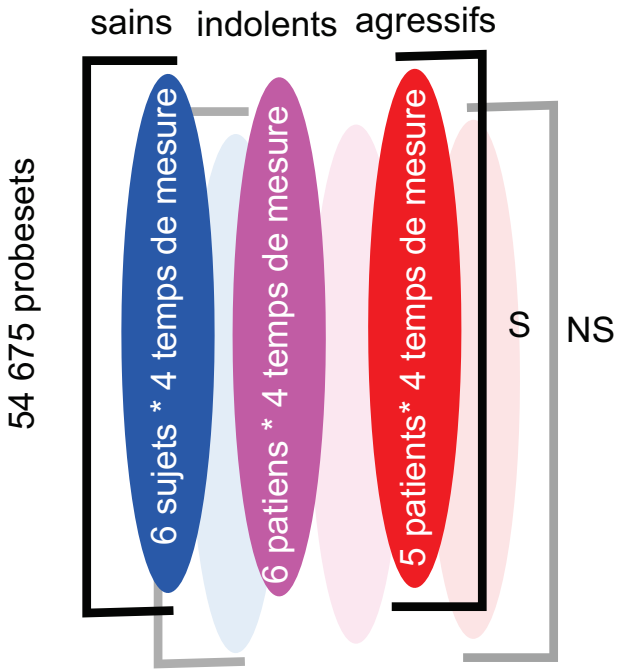


FIGURE 3.2 – *Image venant d'une micropuce à ADN. Chaque point correspond à un probeset particulier.*

réel. De cette manière il est possible d'approximer le nombre de faux-positifs.

La technique de mesure est une technique colorimétrique. Chaque probeset est représenté par plusieurs sondes. Plus le nombre de sondes s'étant appariées avec un brin d'ADN complémentaire coloré est grand, plus la région sur la plaque correspondante va avoir une intensité lumineuse élevée. Un exemple est donné dans la Figure 3.3.

Cette technique donne des mesures très bruitées, ce qui rend l'analyse délicate. Depuis quelques années, une nouvelle technique basée sur le séquençage haut débit permet d'obtenir des résultats plus précis (Next-Generation Sequencing, ou NGS).

Les biopuces utilisées dans cette étude sont de type Affymetrix.

3.2.2 Analyse succincte du jeu de données

Nous nous proposons maintenant de faire une rapide analyse du jeu de données. L'analyse de ce jeu de données est l'objet principal de cette thèse ; l'analyse que nous voulons faire ici doit simplement donner aux lecteurs certains éléments clefs, qui bien que répétés dans la suite de cette thèse, qui leur permettront une lecture plus aisée.

Tout d'abord, nous avons sélectionné les gènes différentiellement exprimés entre l'état stimulé et l'état non stimulé (sans discrimination sur l'état de l'individu) (Smyth 2005). Cela nous permet de sélectionner environ neuf-mille probesets. Une première classification, sur l'ensemble des expressions différentielles de gènes sélectionnés, est faite à l'aide de l'algorithme des k-

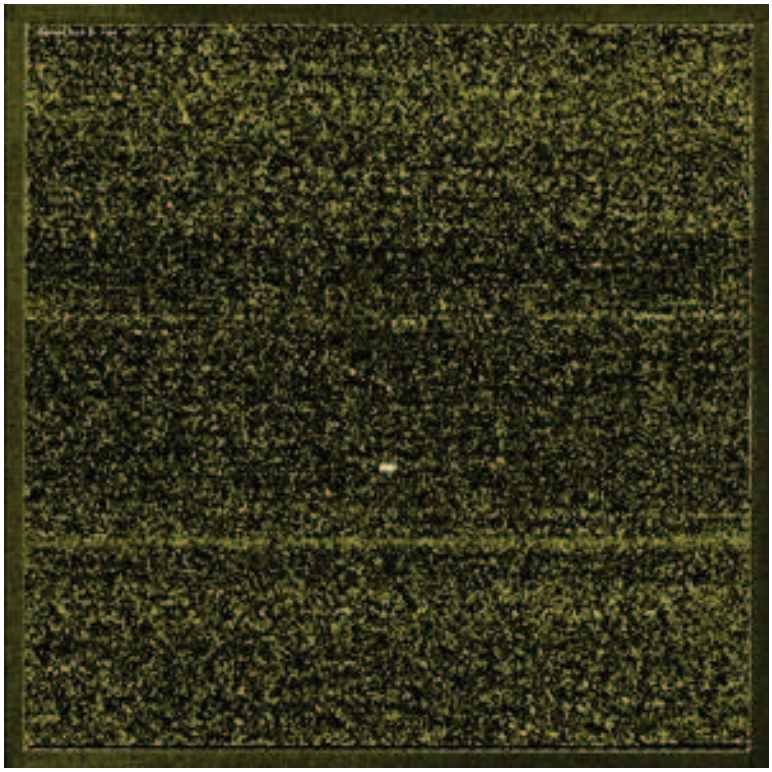


FIGURE 3.3 – *Image venant d'une micropuce à ADN. Chaque point correspond à un probe-set particulier.*

means (Figure 3.4).

Cette analyse nous montre plusieurs choses : d'une part, les expressions différentielles de gènes peuvent être positives ou négatives. Cela implique que certains gènes sont plus exprimés lorsque le BCR n'est pas stimulé, tandis que d'autres voient leur expression diminuer. Ensuite, il n'y a pas de cluster dans lequel les expressions différentielles seraient positives pour un des trois états (sain, malade avec la version agressive de la maladie, malade avec la version indolente de la maladie) et négatives pour les autres (ou inversement). Enfin, l'expression différentielle absolue des individus ayant la forme agressive de la LLC est globalement plus élevée.

Nous nous sommes ensuite interrogés sur les patterns d'expression temporelle que pouvaient avoir les probesets. Pour ce faire, nous avons fait une classification par les k-means en mélangeant l'ensemble des patients (Figure 3.5). Une telle analyse sera faite plus en détails dans les chapitres suivants. Ce que nous voulons souligner ici, c'est les patterns temporels particuliers ; en effet, beaucoup d'expressions différentielles de gènes ont ce que nous appellerons un pattern de pic. Les gènes qui ont un tel pattern ont une expression différentielle nulle, sauf à un temps (voire deux) particulier. Cela nous conduira à la notion de cluster temporel, où les gènes du cluster temporel k seront les gènes ayant un pic au $k$ème temps. Et finalement, cela nous conduira à la notion de réseau en cascade (Figure 3.6) dans lequel des contraintes temporelles particulières seront mises en place. En particulier, les gènes d'un cluster temporel k ne pourront agir que sur les gènes des clus-

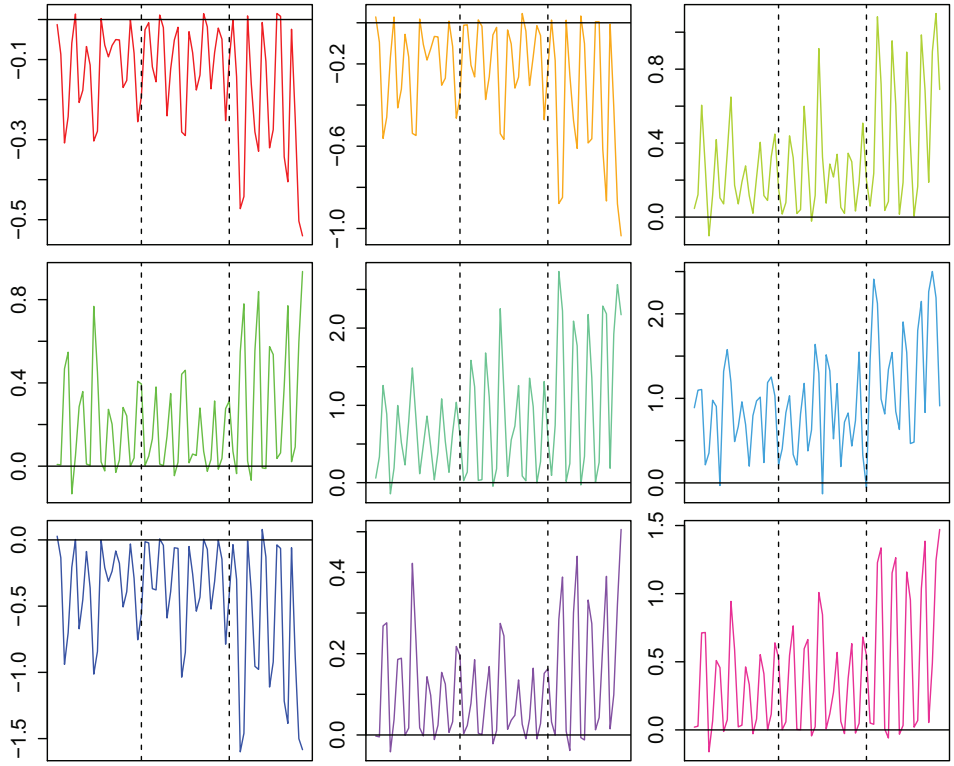


FIGURE 3.4 – *Classification des expressions de gènes retenus après sélection. Chaque graphique est divisé en trois parties. La partie de droite correspond aux expressions des individus sains, celle du milieu aux patients atteints de la forme indolente de la LLC et celle de gauche aux patients atteints de la forme agressive.*

ters temporels k', avec $k' > k$.

Nous allons terminer en essayant de répondre à la question suivante : les expressions de gènes mesurées peuvent-elles discriminer entre temps tardifs (210 et 390 minutes) et temps précoces (60 et 90 minutes) ? Et plus important : les expressions de gènes peuvent-elles permettre de discriminer les individus sains, ceux atteints de la LLC dans sa version indolente et ceux atteints dans la version agressive ?

Pour cela, nous avons décidé d'utiliser une variante de la régression PLS ("partial least square" ou régression des moindres carrés partiels en français) (Wold 1985). La PLS permet de trouver des composantes latentes orthogonales provenant de l'espace vectoriel engendré par les variables explicatives telles que la covariance entre ces variables latentes et la variable réponse soit maximale. Autrement dit, ces variables latentes ont pour double objectif de résumer au mieux l'information contenue dans les variables explicatives et d'être corrélées au mieux à la variable réponse. C'est donc une méthode hybride entre l'analyse en composantes principales et la régression linéaire. Dans notre cas, nous utiliserons la PLS en analyse discriminante (PLS-DA) (Pérez-Enciso et Tenenhaus 2003) qui permet d'utiliser une variable réponse qui est un facteur à plusieurs niveaux. Dans l'approche de PLS-DA que nous avons utilisée, le nombre de variables utilisées pour construire les variables latentes est limité. Ce dernier est choisi de sorte à minimiser l'erreur de prédiction (calcul réalisé par validation croisée) (Lê Cao *et al.* 2011). Les

graphiques présentés dans les Figures 3.7, 3.8 et 3.9 sont les projections des individus sur l'espace vectoriel engendré par les deux premières composantes latentes.

Dans la première analyse (Figure 3.7) la variable réponse est un facteur contenant 6 niveaux, chacun correspondant à un état de l'individu (sain, LLC indolente, LLC agressive) et à un ensemble de temps donnés (précoce : 60 et 90 minutes, tardif : 210 et 390 minutes). Deux éléments sont à noter : les expressions de gènes permettent facilement de distinguer les temps précoces des temps tardifs (axe gauche droite) et parmi les temps tardifs, les trois états des individus sont bien séparés. En revanche, cette analyse ne permet pas de distinguer les trois états des individus parmi les temps précoces (même sur les plans des variables latentes supérieures, figures non montrées). Cela nous conduit à séparer notre analyse en deux, entre temps précoces d'un côté et temps tardifs d'un autre côté.

L'analyse pour les temps précoces présente des résultats intéressants (Figure 3.8). Tout d'abord, nous voyons que nous parvenons parfaitement à discriminer les trois types d'individus. Ensuite, il faut noter que seuls 40 variables (ici, donc, des probesets) ont été retenues par les composantes latentes. Une analyse des 80 probesets retenus pour construire le plan présenté en Figure 3.8 permet une analyse de la signification biologique de la différence entre les différents états d'individus. Nous donnons ici la liste de ceux ayant une identification connue (Tableau 3.1).

La liste de ces gènes donne un aperçu de certains gènes qui seront particulièrement étudiés dans le chapitre suivant. Par exemple, les deux gènes ayant 3 probesets associés seront des hubs dans la reconstruction de réseau. Par ailleurs, le gène DUSP1 est à remarquer également, car une attention toute particulière lui sera accordée au Chapitre 5. Mais globalement, cette liste de gènes ne donne pas d'indication suffisante. Il faut alors analyser les fonctions biologiques de ces gènes, et regarder si la liste présentée n'a pas une surreprésentation de gènes ayant une certaine fonction. Nous utilisons à cette fin l'outil en ligne DAVID (Da Wei Huang et Lempicki 2008) [1]. Nous montrons ici les fonctions biologiques associées significativement aux probesets retenus :

— noyau (p-valeur : 0.03),
— phosphoprotéine (p-valeur 0.03),
— liaison à l'ARN (p-valeur 0.035),
— facteur transcriptionnel (p-valeur : 0.049).

La même analyse est appliquée aux temps tardifs (Figure 3.9). Nous remarquons également que cette analyse permet de discriminer les individus des trois types. Seulement, ici seuls 15 probesets ont été retenus par variable latente. Nous en donnons également la liste (Tableau 3.2). De la même manière, nous étudions les fonctions biologiques associées à ces gènes. Nous trouvons les fonctions suivantes :

1. `http://david.abcc.ncifcrf.gov/`

Nom du gène	Nombre de probesets	Nom du gène	Nombre de probesets
ALYREF	1	MCC	1
ARL4C	1	NFE2L1	1
ATXN7	1	OSBPL10	1
AVEN	1	PCK2	1
CBR4	1	PHF21A	1
CCDC181	1	RASGRP2	1
CD24	1	RBM28	1
CDPF1	1	REL	1
CHD2	1	RGCC	1
CHRNA1	1	RPL36A	1
COMMD3	1	RPP25	1
COX5A	1	SAMSN1	1
DDX50	1	SCRIB	1
DNAJC11	1	SEPHS2	1
DUSP1	1	SLAMF1	1
EGR1	3	SRGAP2B	1
EGR4	1	SRGN	1
EIF4A1	1	STK24	1
EIF5	1	TAF9	1
EPC1	1	THRAP3	1
EXOSC6	1	TMF1	1
FAM46C	1	TRA2B	1
ITPR1-AS1	1	UBAC1	1
JUN	3	USP21	1
KLF2	1	ZNF182	1
LONRF1	1	ZNF224	1

TABLE 3.1 – *Liste de gènes correspondant aux temps précoces.*

— régulation positive de l'activation des cellules B,
— régulation positive de la prolifération des cellules,
— régulation positive de l'activation des lymphocytes,
— régulation positive de l'activation des leucocytes.

Sans rentrer dans les détails, nous voyons donc que les gènes d'expressions tardives correspondent à des fonctions auxquelles nous pouvons nous attendre dans le cadre de la LLC.

Nom du gène	Nombre de probesets
ADARB1	1
ASAH1	1
BEND3	1
C19orf48	1
CRTC3	1
CXXC5	2
ELL3	2
GBP2	2
KLK2	1
NETO1	1
ODC1	1
SNX5	1
SUSD3	1
SYK	2
TCF3	1
TESPA1	1

TABLE 3.2 – *Liste de gènes correspondant aux temps tardifs.*

3.3 CONCLUSION

Nous avons présenté dans ce chapitre notre modèle biologique et le jeu de données en résultant. Une première analyse de ce dernier nous a permis de montrer que quelques dizaines de gènes sont suffisants dans l'objectif de discriminer les différents types d'individu (sain, malade avec la version agressive de la LLC, malade avec la version indolente de la LLC). Cette étude préliminaire appelle donc à des études plus complexes. Notre travail va consister à utiliser les outils statistiques issus de la théorie de la biologie des systèmes complexes pour obtenir de nouvelles informations. En particulier, nous chercherons dans un premier temps à découvrir la structure de ce système, ainsi que sa dynamique. Se posera ensuite la question de la contrôlabilité d'un tel système (Chapitre 5) avant l'ébauche d'un travail sur la possibilité d'une modification orientée de ce dernier (Chapitre 9).

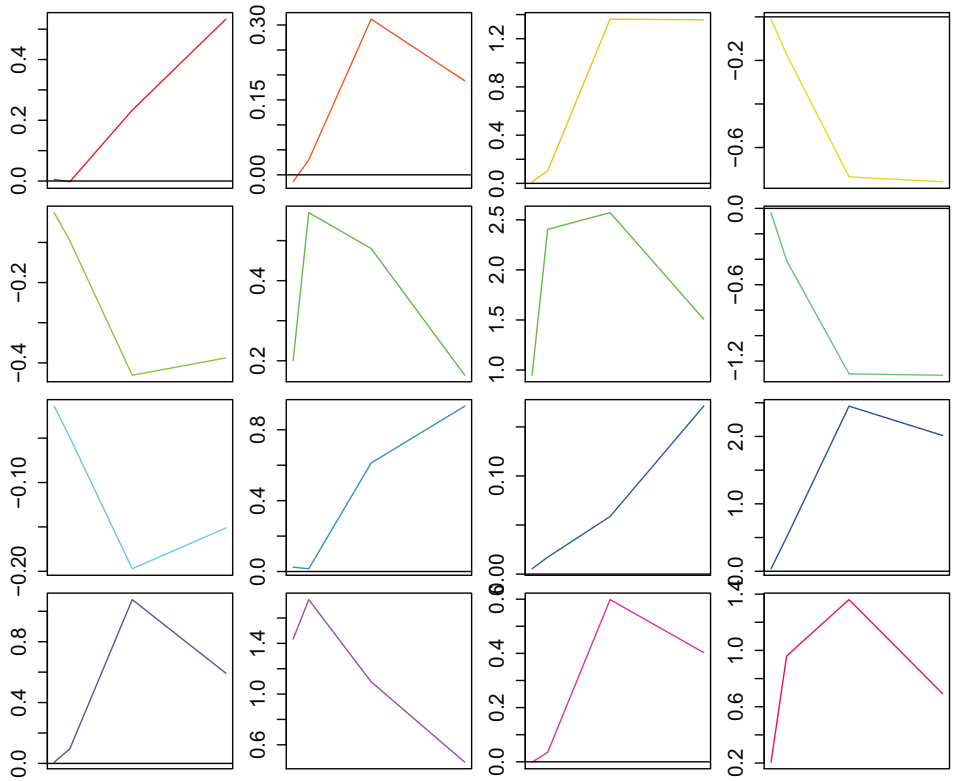


FIGURE 3.5 – *Les différents patterns temporels dans les expressions différentielles des gènes.*

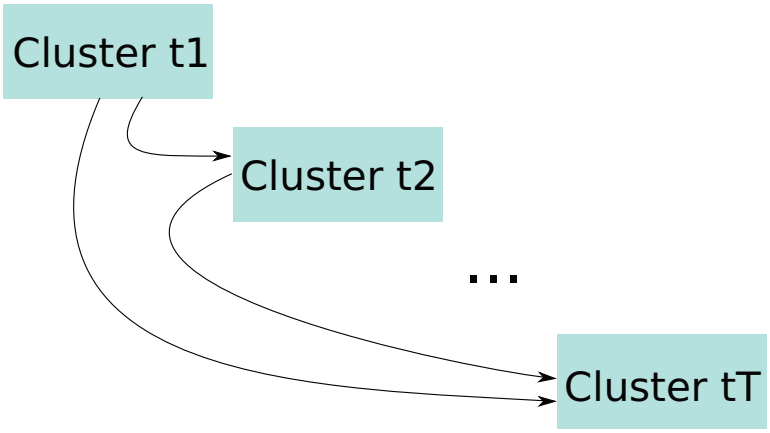


FIGURE 3.6 – *Illustration de la notion de réseau en cascade.*

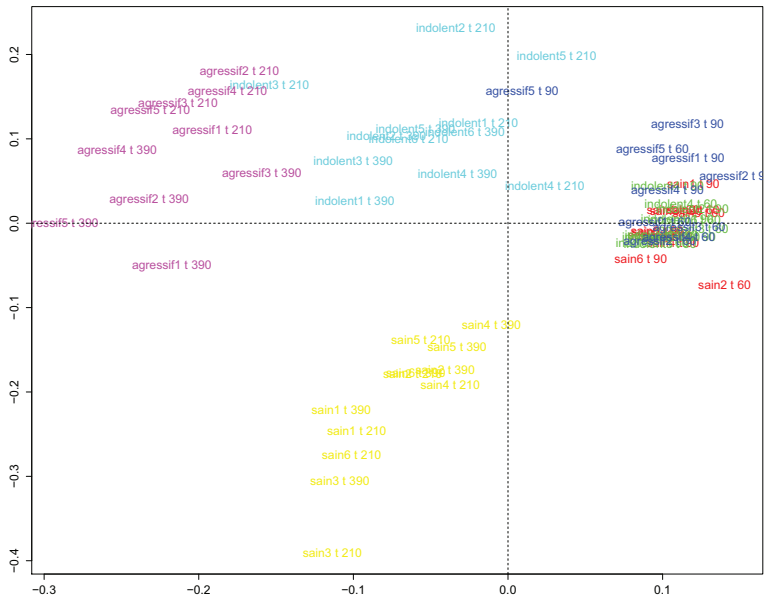


FIGURE 3.7 – *PLS-DA parcimonieuse : les expressions temporelles précoces ne permettent pas de distinguer les différents patients, contrairement aux expressions des temps tardifs.*

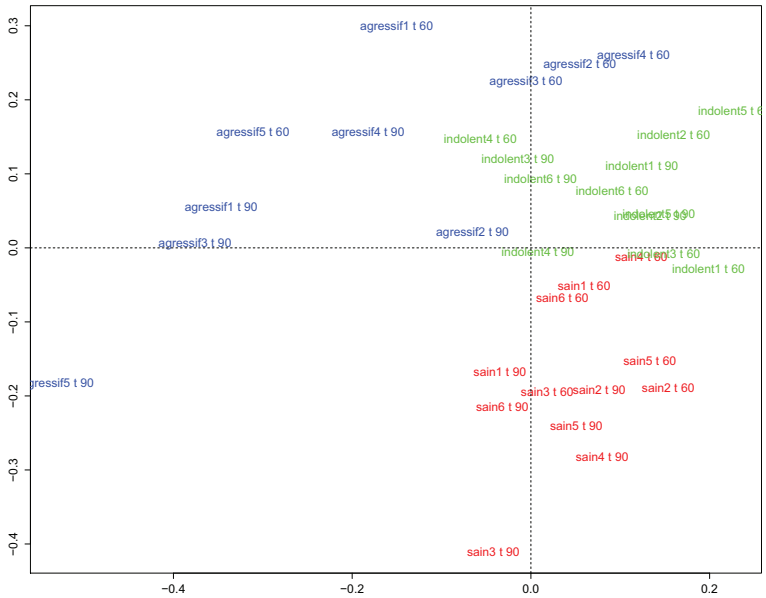


FIGURE 3.8 – *PLS-DA parcimonieuse : les expressions précoces, analysées à part, permettent de discriminer les différents individus.*

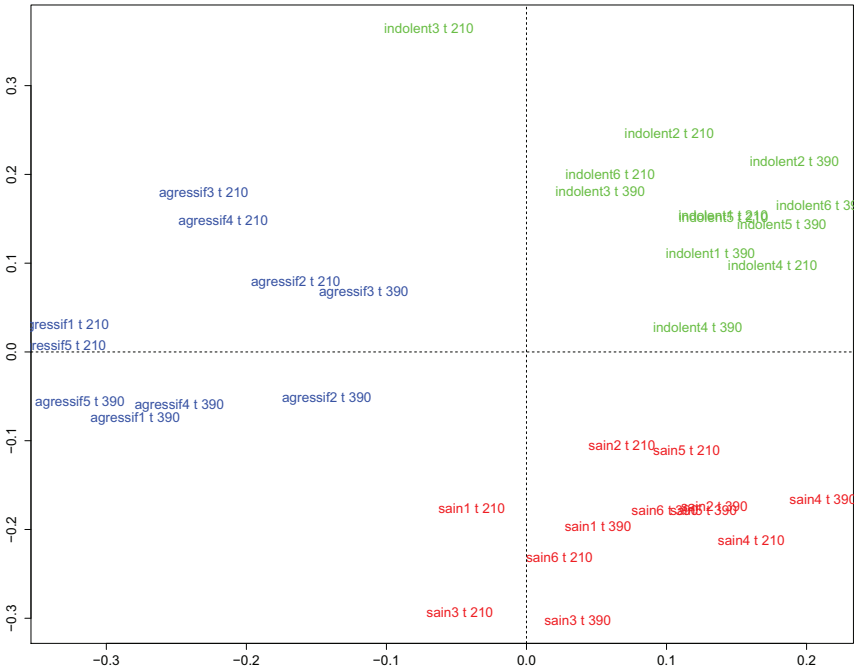


FIGURE 3.9 – *PLS-DA parcimonieuse : les expressions tardives, analysées à part, permettent de discriminer les différents individus.*

Deuxième partie

Vers une modulation orientée dans un programme génique

Modélisation en cascade d'un réseau de gène

4

Dans la première partie nous avons détaillé en trois chapitres les notions et les concepts utiles - tant d'un point de vue biologique que d'un point de vue mathématique et statistique - à la compréhension de cette thèse. Dans le premier chapitre nous avons défini le cadre général de la biologie des systèmes. Nous en avons profité pour nous attarder sur la théorie des réseaux, lesquels peuvent être vus comme une représentation utile de ces systèmes. Dans le deuxième chapitre de la partie introductive nous avons présenté les méthodes statistiques de sélection de variables dans la régression linéaire qui seront utilisés dans la suite pour l'inférence de réseaux ; en particulier, nous nous sommes attardés sur les propriétés de la régression Lasso. La sélection de variables est un outil puissant dans la reconstruction de réseaux de régulation génique, puisque ces derniers sont supposés jouir de la propriété de parcimonie (un gène est régulé par un nombre limité d'autres gènes). Dans le troisième chapitre nous avons présenté notre modèle biologique et du jeu de données en résultant. Un premier examen rapide nous a permis de déterminer les caractéristiques essentielles de ce jeu de données ; nous y reviendrons dans un instant.

Le modèle biologique présenté dans le Chapitre 3 a cela de particulier que la stimulation initiale est pulsée et non pas continue. Autrement dit, la stimulation utilisée est très intense sur une brève période temporelle. Comme le montre la Figure 3.5 cela a des conséquences sur le pattern des expressions de gènes. Nous n'observons pas d'expression de gène qui soit cyclique mais plutôt des pics d'expressions différentielles qui interviennent à des temps d'observation bien précis. Cela nous a permis de poser le concept dans lequel chaque gène est associé à un temps d'observation. L'idée, assez intuitive, est alors de considérer qu'un gène d'un temps donné ne peut agir sur un autre gène que si ce dernier est associé à un temps ultérieur. Cela est représenté dans la Figure 3.6. Ce modèle est bien évidemment une simplification de la réalité que nous affinerons par la suite.

4.1 Modélisation en cascade

Pour établir cette modélisation d'un réseau en cascade nous sommes partis d'une approche existante publiée dans une thèse (Kemper 2006). Cependant les travaux présentés dans cette thèse sont incomplets et nous avons

dû les compléter. Nous présentons dans un premier temps le modèle initial avant de mettre en lumière les modifications que nous lui avons apporté.

4.1.1 Clustering

Il a été conjecturé que les gènes les plus importants, en particulier les facteurs transcriptionnels, ont une activité intensifiée à des moments précis. L'idée est alors de mettre dans un même cluster les gènes qui ont leur pic d'activité au même moment. Puisque nous décrivons par des temps discrets un processus continu, le choix des instants de mesure est essentiel[1]. Nous supposons que nous disposons d'un vecteur d'observations $\boldsymbol{x}$ d'une variable aléatoire X. Ces observations proviennent de N gènes, de P patients et de T instants de mesure notés $\{t_1, ..., t_T\}$. Nous désignerons par x_{npt_k} l'expression du gène n pour le patient p au temps t_k. Nous définissons également $\boldsymbol{x}_{n..}$ le vecteur des observations pour le gène n.

Nous définissons alors $T+1$ clusters :

— Un cluster par temps de mesure (du cluster 1 jusqu'au cluster T), contenant chacun les gènes qui s'expriment de manière prépondérante et positive à l'instant considéré (ie un gène du cluster t_k avec $1 \leqslant k \leqslant T$ va avoir une expression prépondérante à l'instant t_k). En effet, nous ne nous intéresserons ici qu'aux gènes étant influencés positivement par la stimulation initiale. Nous appellerons désormais un gène appartenant à un cluster t_k avec $1 \leqslant k \leqslant T$ un t_k-gène ou un gène du temps t_k.
— Le $T+1^{eme}$ cluster correspond aux gènes dont l'expression ne varie pas au cours du temps, c'est-à-dire aux gènes qui ne sont pas influencés par la stimulation, ou qui sont influencés négativement, comme le montre le schéma dans la Figure 4.1

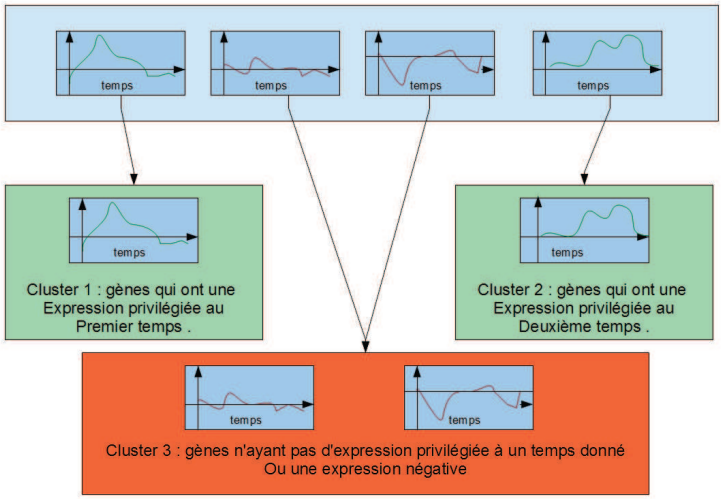


FIGURE 4.1 – *Exemple d'application de l'algorithme de clustering avec quatre gènes et deux temps de mesure (les répétitions liées aux patients ne sont pas représentées ici).*

1. Ce choix a été fait dans une étude préliminaire

La méthode de clustering choisie repose sur un modèle de mélange que nous estimerons par un algorithme EM (expectation-maximization). Formellement, nous allons supposer que la loi de la variable aléatoire X est un mélange fini de lois exponentielles et normales. Chaque composante de ce mélange correspondra ensuite à un cluster donné. Pour la partie de clustering (et seulement pour cette partie), les temps de mesure sont supposés indépendants ; en revanche les patients sont toujours considérés comme indépendants tout au long de l'analyse.

Puisque l'objectif est de classifier les gènes en fonction du moment de leur plus forte activité, les patients et les temps de mesure sont considérés comme des répétitions pour un gène donné. Par conséquent, nous supposons que la vraisemblance à maximiser est de la forme suivante :

$$L(\Theta|\boldsymbol{x}) = \prod_{n=1}^{N} \sum_{m=1}^{T+1} p_m(\boldsymbol{x}_{n..}|\Theta_m)\pi_m \tag{4.1}$$

où $\Theta = \bigcup_{m=1}^{T+1} \Theta_m$ avec Θ_m représentant l'ensemble des paramètres pour la densité du cluster m, et π_m qui représente la probabilité d'être dans le cluster m, ou encore, la proportion d'observations dans le cluster m, et $\boldsymbol{x}_{n..}$ le vecteur d'expression du gène n pour tous les patients et tous les temps. Comme les patients et les temps de mesure sont supposés être des répétitions indépendantes pour un gène n donné, nous pouvons poser :

$$p_m(\boldsymbol{x}_{n..}|\Theta_m) = \prod_{p=1}^{P} \prod_{i=1}^{T} p_m(x_{npt_i}|\Theta_m). \tag{4.2}$$

Afin de modéliser les pics d'activité des gènes à des moments localisés dans le temps, nous posons :

$$p_m(x_{npt_i}|\Theta_m) = \begin{cases} \lambda_m \exp(-\lambda_m x_{npt_i}) & ; \quad x_{npt_i} > 0; 1 \leqslant i \leqslant 4; i = m \\ 0 & ; \quad x_{npt_i} \leqslant 0; 1 \leqslant i \leqslant 4; i = m \\ \dfrac{\lambda_{m_{t_i}^+}}{2} \exp(-\lambda_{m_{t_i}^+} x_{npt_i}) & ; \quad x_{npt_i} > 0; 1 \leqslant i \leqslant 4; i \neq m \\ \dfrac{\lambda_{t_i^-}}{2} \exp(-\lambda_{t_i^-} x_{npt_i}) & ; \quad x_{npt_i} \leqslant 0; 1 \leqslant i \leqslant 4; i \neq m \\ \dfrac{1}{\sqrt{2\pi\sigma_{t_i}^2}} \exp\left(-\dfrac{1}{2}\dfrac{x_{npt_i}^2}{\sigma_{t_i}^2}\right) & ; \quad 1 \leqslant i \leqslant 4; m = 5 \end{cases} . \tag{4.3}$$

Comme il a déjà été précisé, le cluster m, $m \leqslant T$, contient les gènes dont le pic est le plus sensible au temps t_m ; c'est pourquoi nous modélisons l'expression d'un gène du cluster m, $m \leqslant T$, au temps t_m par une loi exponentielle unilatérale. Les autres temps sont modélisés à l'aide d'un mélange d'une loi exponentielle positive et négative. La loi exponentielle a été préférée à d'autres distributions, en particulier à la loi normale, car sa lourde queue modélise mieux la dispersion des résultats. Notre intérêt se portant uniquement sur les expressions positives, un paramètre commun $\lambda_{t_i^-}$ a été

choisi pour chaque point de temps t_i. Enfin, une loi normale centrée de variance $\sigma^2_{t_i}$ modélise les gènes qui n'ont pas de pic d'activité particulier.

Nous remarquons tout de suite le défaut d'une telle modélisation. En effet, si un gène n appartient au cluster k mais que pour un des patients p nous avons $x_{npt_k} \leqslant 0$, nous rejetteront à tort (avec probabilité 1) ce gène n du cluster k. Compte tenu de l'amplitude élevée du bruit dans les données recueillies par microarrays, cette situation est fréquente. Pour pallier ce problème, la règle de flexibilité suivante est posée :

Si pour un gène donné un patient et un seul empêche l'attribution à un cluster, ce patient est ignoré pour ladite attribution.

Enfin, nous attribuons les gènes aux clusters, en considérant la probabilité *a a posteriori* maximale ("MAP criterion") :

$$k_n = \underset{k\in 1\ldots N+1}{\arg\max} \frac{\pi_k \left(\prod_{p=1}^{P} \prod_{i=1}^{T} p_k(x_{npt_i}|\Theta_k) \right)}{\sum_{m=1}^{T+1} \pi_m \left(\prod_{p=1}^{P} \prod_{i=1}^{T} p_m(x_{npt_i}|\Theta_m) \right)} \tag{4.4}$$

k_n se lisant "le gène n appartient au cluster k".

En remarquant que le dénominateur ne dépend pas de k, nous pouvons simplifier l'expression de k_n de la manière suivante :

$$k_n = \underset{k}{\arg\max} \left[\pi_k \left(\prod_{p=1}^{P} \prod_{i=1}^{T} p_k(x_{npt_i}|\Theta_k) \right) \right]. \tag{4.5}$$

Maintenant que le modèle est posé, nous allons donner les itérations de l'algorithme EM. Les sous-parties ci-dessous sont les détails calculatoires de l'algorithme. Le lecteur peut passer directement à la section 4.1.2 sans altérer la compréhension du travail.

Mise en place de l'algorithme EM

L'algorithme EM permet de maximiser une vraisemblance dans laquelle certaines variables sont inconnues. Dans notre cas, ces variables inconnues représentent le cluster d'un gène donné. Pour formaliser cette idée, nous allons introduire les vecteurs aléatoires latents $\{\boldsymbol{Z}_1, .., \boldsymbol{Z}_N\}$ tels que :

$$\boldsymbol{Z}_n = \begin{pmatrix} Z_n^{(1)} \\ \vdots \\ Z_n^{(T+1)} \end{pmatrix}.$$

Nous imposons alors les contraintes suivantes à ces variables latentes :

$$\begin{cases} Z_n^{(m)} \in \{0,1\} \\ \mathbb{P}(Z_n^{(m)} = 1) = \pi_m \\ \sum_{m=1}^{T+1} Z_n^{(m)} = 1 \end{cases}.$$

En d'autres mots, $\boldsymbol{Z}_n$ est un vecteur contenant T zéros et un unique un. La position de ce "un" donne l'attribution du gène n à un cluster. Nous pouvons alors poser :

$$p(\boldsymbol{x}_{n..}|\Theta; \boldsymbol{Z}_n = \boldsymbol{z}_n) = \prod_{m=1}^{T+1} \left(p_m(\boldsymbol{x}_{n..}|\Theta_m)\right)^{z_n^{(m)}}$$

où $\boldsymbol{z}_n$ est la réalisation du vecteur aléatoire $\boldsymbol{Z}_n$ et $z_n^{(m)}$ est la réalisation de la variable aléatoire $Z_n^{(m)}$.

La densité complète s'écrit alors de la forme suivante :

$$p(x_{n..}, \boldsymbol{z}_n|\Theta) = \prod_{m=1}^{T+1} \left(\pi_m p_m(x_{n..}|\Theta_m)\right)^{z_n^{(m)}}.$$

Nous allons maintenant calculer la log-vraisemblance des données complètes :

$$\begin{aligned}
\mathcal{L}_\theta^c(x,z) &= \log\left(\prod_{n=1}^{N}\prod_{m=1}^{T+1}\left(\pi_m p_m(x_{n..}|\Theta_m)\right)\right)^{z_n^{(m)}}\Bigg) \\
&= \sum_{n=1}^{N}\sum_{m=1}^{T+1}\log\left(\left(\pi_m p_m(x_{n..}|\Theta_m)\right)^{z_n^{(m)}}\right) \\
&= \sum_{n=1}^{N}\sum_{m=1}^{T+1}\log\left(\left(\pi_m \prod_{p=1}^{P}\prod_{i=1}^{T} p_m(x_{npt_i}|\Theta_m)\right)^{z_n^{(m)}}\right) \\
&= \sum_{n=1}^{N}\sum_{m=1}^{T+1} z_n^{(m)}\log\left(\pi_m \prod_{p=1}^{P}\prod_{i=1}^{T} p_m(x_{npt_i}|\Theta_m)\right) \\
&= \sum_{n=1}^{N}\sum_{m=1}^{T+1} z_n^{(m)}\left(\log(\pi_m) + \sum_{p=1}^{P}\sum_{i=1}^{T}\log\left(p_m(x_{npt_i}|\Theta_m)\right)\right)
\end{aligned}$$

où nous avons posé la notation $\mathcal{L}_\theta^c(x,z) \triangleq \mathcal{L}(\Theta|\boldsymbol{x}_{1..}, ..., \boldsymbol{x}_{n..}, \boldsymbol{z}_1, .., \boldsymbol{z}_n)$.

Etape E (Expectation step)

Cette étape consiste à calculer l'espérance de log-vraisemblance des données complètes en gardant les vecteurs $\boldsymbol{Z}_1, ..., \boldsymbol{Z}_N$ aléatoires, en supposant que nous disposons d'une première estimation $\Theta^\star$ des paramètres. Nous posons alors :

$$Q(\Theta|\Theta^\star) = \mathbb{E}_{\boldsymbol{Z}_1,\dots,\boldsymbol{Z}_N}\left(\sum_{n=1}^{N}\sum_{m=1}^{T+1} Z_n^{(m)}\left(\log(\pi_m) + \sum_{p=1}^{P}\sum_{i=1}^{T}\log\left(p_m(X_{npt_i}|\Theta_m)\right)\right)\middle|\,\Theta^\star, \boldsymbol{x}_{n..}\right)$$
$$= \sum_{n=1}^{N}\sum_{m=1}^{T+1}\mathbb{E}_{\boldsymbol{Z}_1,\dots,\boldsymbol{Z}_N}\left(Z_n^{(m)}\middle|\,\Theta^\star, \boldsymbol{x}_{n..}\right)\left(\log(\pi_m) + \sum_{p=1}^{P}\sum_{i=1}^{T}\log\left(p_m(x_{npt_i}|\Theta_m)\right)\right).$$

Nous rappelons que le but est de maximiser cette dernière fonction en Θ, en connaissant l'estimation $\Theta^\star$ de l'étape précédente. Par le théorème de Bayes, nous écrivons :

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{Z}_1,\dots,\boldsymbol{Z}_n}\left(Z_n^{(m)}\middle|\,\Theta^\star, \boldsymbol{x}_{n..}\right) &= \mathbb{P}\left(Z_n^{(m)} = 1\middle|\,\Theta^\star, \boldsymbol{x}_{n..}\right)\\
&= \frac{\mathbb{P}\left(Z_n^{(m)} = 1\middle|\,\Theta^\star\right) p(\boldsymbol{x}_{n..}|Z_n^{(m)} = 1, \Theta^\star)}{\sum_{m=1}^{T+1}\mathbb{P}\left(Z_n^{(m)} = 1\middle|\,\Theta^\star\right) p(\boldsymbol{x}_{n..}|Z_n^{(m)} = 1, \Theta^\star)}\\
&= \frac{\mathbb{P}\left(Z_n^{(m)} = 1\middle|\,\Theta^\star\right) p_m(\boldsymbol{x}_{n..}|\Theta^\star)}{\sum_{m=1}^{T+1}\mathbb{P}\left(Z_n^{(m)} = 1\middle|\,\Theta^\star\right) p_m(\boldsymbol{x}_{n..}|\Theta^\star)}\\
&= \frac{\pi_m^\star\ p_m(\boldsymbol{x}_{n..}|\Theta^\star)}{\sum_{m=1}^{T+1}\pi_m^\star\ p_m(\boldsymbol{x}_{n..}|\Theta^\star)}\\
&= \Pi_{nm}.
\end{aligned}$$

Nous avons posé $\pi_m^\star$ la proportion du cluster m en supposant que le vrai paramètre est $\Theta^\star$. Nous interprétons Π_{nm} comme la probabilité pour le gène n d'être dans le cluster m, étant donnés les observations et les paramètres $\Theta^\star$. Nous avons alors :

$$Q(\Theta|\Theta^\star) = \sum_{n=1}^{N}\sum_{m=1}^{T+1}\Pi_{nm}\left(\log(\pi_m) + \sum_{p=1}^{P}\sum_{i=1}^{T}\log\left(p_m(x_{npt_i}|\Theta_m)\right)\right). \quad (4.6)$$

Nous pouvons maintenant passer à l'étape de maximisation.

Etape M (Maximization step)

Il s'agit maintenant de trouver un ensemble de paramètres Θ qui maximise la fonction $Q(\Theta|\Theta^\star)$. Les paramètres ainsi trouvés seront appelés paramètres mis à jour.

Nous allons commencer par mettre à jour $\sigma_{t_i}^2$ pour un *i fixé* dans $\{1, ..., T\}$. Comme $\sigma_{t_i}^2$ n'intervient que pour le dernier cluster $T+1$, la fonction f_1 à maximiser sera :

$$
\begin{aligned}
f_1(\sigma_{t_i}^2) &= \sum_{n=1}^{N} \Pi_{n(T+1)} \left(\sum_{p=1}^{P} \log \left(p_{T+1}(x_{npt_i} | \Theta_{T+1}) \right) \right) \\
&= \sum_{n=1}^{N} \sum_{p=1}^{P} \Pi_{n(T+1)} \log \left(p_{T+1}(x_{npt_i} | \Theta_{T+1}) \right) \\
&= \sum_{n=1}^{N} \sum_{p=1}^{P} \Pi_{n(T+1)} \log \left(\frac{1}{\sqrt{2\pi\sigma_{t_i}^2}} \exp \left(-\frac{1}{2} \frac{x_{npt_i}^2}{\sigma_{t_i}^2} \right) \right) \\
&= \sum_{n=1}^{N} \sum_{p=1}^{P} \Pi_{n(T+1)} \left(-\frac{\log(2\pi\sigma_{t_i}^2)}{2} - \left(\frac{1}{2} \frac{x_{npt_i}^2}{\sigma_{t_i}^2} \right) \right).
\end{aligned}
$$

Il suffit maintenant d'annuler la dérivée :

$$
\begin{aligned}
& \frac{df_1}{d\sigma_{t_i}^2} = 0 \\
\Leftrightarrow & \sum_{n=1}^{N} \sum_{p=1}^{P} \Pi_{n(T+1)} \left(-\frac{1}{2\sigma_{t_i}^2} + \frac{1}{2} \frac{x_{npt_i}^2}{\sigma_{t_i}^4} \right) = 0 \\
\Leftrightarrow & \sum_{n=1}^{N} \sum_{p=1}^{P} \Pi_{n(T+1)} \left(-\sigma_{t_i}^2 + x_{npt_i}^2 \right) = 0 \\
\Leftrightarrow & \sigma_{t_i}^2 = \frac{\sum_{i=1}^{N} \sum_{p=1}^{P} \Pi_{n(T+1)} x_{npt_i}^2}{\sum_{n=1}^{N} \sum_{p=1}^{P} \Pi_{n(T+1)}}.
\end{aligned}
$$

Nous avons donc trouvé la formule de mise à jour pour la variance de la loi normale. Passons maintenant au paramètre λ_m, pour un m *fixé* dans $\{1, ..., T\}$. Nous remarquons que λ_m n'intervient que lorsque t_i est tel que $i = m$; nous pouvons donc poser la fonction f_2 :

$$
\begin{aligned}
f_2(\lambda_m) &= \sum_{\substack{n=1 \\ x_{npt_m} > 0}}^{N} \Pi_{nm} \left(\sum_{p=1}^{P} \log \left(p_m(x_{npt_m} | \Theta) \right) \right) \\
&= \sum_{\substack{n=1 \\ x_{npt_m} > 0}}^{N} \sum_{p=1}^{P} \Pi_{nm} \left(\log \left(p_m(x_{npt_m} | \Theta) \right) \right) \\
&= \sum_{\substack{n=1 \\ x_{npt_m} > 0}}^{N} \sum_{p=1}^{P} \Pi_{nm} \left(\log(\lambda_m) - \lambda_m x_{npt_m} \right).
\end{aligned}
$$

Il s'agit comme avant de maximiser cette fonction, en annulant la dérivée :

$$\begin{aligned}
&\frac{df}{d\lambda_m} = 0 \\
\Leftrightarrow &\sum_{\substack{n=1 \\ x_{npt_m}>0}}^{N} \sum_{p=1}^{P} \Pi_{nm} \left(\frac{1}{\lambda_m} - x_{npt_m} \right) = 0 \\
\Leftrightarrow \lambda_m = &\frac{\sum_{p=1}^{P} \sum_{\substack{n=1 \\ x_{npt_m}>0}}^{N} \Pi_{nm}}{\sum_{p=1}^{P} \sum_{\substack{n=1 \\ x_{npt_m}>0}}^{N} \Pi_{nm} x_{npt_m}}.
\end{aligned}$$

Des calculs similaires permettent de trouver la mise à jour pour $\lambda_{m_{t_i}^+}$ avec $i \neq m$ *fixés* :

$$\lambda_{m_{t_i}^+} = \frac{\sum_{p=1}^{P} \sum_{\substack{n=1 \\ x_{npt_i}>0}}^{N} \Pi_{nm}}{\sum_{p=1}^{P} \sum_{\substack{n=1 \\ x_{npt_i}>0}}^{N} \Pi_{nm} x_{npt_i}}.$$

La mise à jour pour le paramètre $\lambda_{t_i^-}$, i *fixé* se fait alors de la manière suivante :

$$\lambda_{t_i^-} = \frac{\sum_{m=1}^{T+1} \sum_{p=1}^{P} \sum_{\substack{n=1 \\ x_{npt_i}\leqslant 0}}^{N} \Pi_{nm}}{\sum_{m=1}^{T+1} \sum_{p=1}^{P} \sum_{\substack{n=1 \\ x_{npt_i}\leqslant 0}}^{N} \Pi_{nm} |x_{npt_i}|}.$$

Il reste maintenant à mettre à jour les proportions π_m. Comme π_m sont des proportions, nous avons naturellement la contrainte :

$$\sum_{m=1}^{T+1} \pi_m = 1 \Leftrightarrow \pi_{T+1} = 1 - \sum_{m=1}^{T} \pi_m. \tag{4.7}$$

Nous utilisons (4.7) dans (4.6) pour définir f_3 :

$$\begin{aligned}
f_3(\pi_1, ..., \pi_T) &= \sum_{n=1}^{N} \sum_{m=1}^{T+1} \Pi_{nm} \left(\log(\pi_m) + \sum_{p=1}^{P} \sum_{i=1}^{T} \log\left(p_m(x_{npt_i} | \Theta_m) \right) \right) \\
&= \sum_{n=1}^{N} \sum_{m=1}^{T+1} \Pi_{nm} \log(\pi_m) + \gamma \\
&= \sum_{n=1}^{N} \left(\sum_{m=1}^{T} \Pi_{nm} \log(\pi_m) + \Pi_{n(T+1)} \log(\pi_{T+1}) \right) + \gamma \\
&= \sum_{n=1}^{N} \left(\sum_{m=1}^{T} \Pi_{nm} \log(\pi_m) + \Pi_{n(T+1)} \log\left(1 - \sum_{m=1}^{T} \pi_m \right) \right) + \gamma.
\end{aligned}$$

Nous calculons la dérivée partielle par rapport à π_k :

$$\begin{aligned}\frac{df_3(\pi_1,...,\pi_T)}{\pi_k} &= \sum_{n=1}^{N}\left(\sum_{m=1}^{T}\Pi_{nm}\frac{d\log(\pi_m)}{d\pi_k}+\Pi_{n(T+1)}\frac{d\log\left(1-\sum_{m=1}^{T}\pi_m\right)}{d\pi_k}\right)\\ &= \sum_{n=1}^{N}\left(\frac{\Pi_{nk}}{\pi_k}-\frac{\Pi_{n(T+1)}}{1-\sum_{m=1}^{T}\pi_m}\right).\end{aligned}$$

Nous avons alors, pour $k=1,...,T$, en annulant la dérivée :

$$\left\{\begin{array}{l}\dfrac{\sum_{n=1}^{N}\Pi_{n1}}{\pi_1}=\dfrac{\sum_{n=1}^{N}\Pi_{n(T+1)}}{1-\sum_{m=1}^{T}\pi_m}\\ ...\\ \dfrac{\sum_{n=1}^{N}\Pi_{nk}}{\pi_k}=\dfrac{\sum_{n=1}^{N}\Pi_{n(T+1)}}{1-\sum_{m=1}^{T}\pi_m}\\ ...\\ \dfrac{\sum_{n=1}^{N}\Pi_{nT}}{\pi_T}=\dfrac{\sum_{n=1}^{N}\Pi_{n(T+1)}}{1-\sum_{m=1}^{T}\pi_m}\end{array}\right. \Leftrightarrow \left\{\begin{array}{l}\pi_1\sum_{n=1}^{N}\left(\Pi_{n1}+\Pi_{n(T+1)}\right)+\sum_{m=2}^{T}\pi_m\sum_{n=1}^{N}\Pi_{n1}=\sum_{n=1}^{N}\Pi_{n1}\\ ...\\ \pi_k\sum_{n=1}^{N}\left(\Pi_{nk}+\Pi_{n(T+1)}\right)+\sum_{m=1,m\neq k}^{T}\pi_m\sum_{n=1}^{N}\Pi_{nk}=\sum_{n=1}^{N}\Pi_{nk}\\ ...\\ \pi_T\sum_{n=1}^{N}\left(\Pi_{nT}+\Pi_{n(T+1)}\right)+\sum_{m=1,m\neq T}^{T}\pi_m\sum_{n=1}^{N}\Pi_{nT}=\sum_{n=1}^{N}\Pi_{nT}\end{array}\right. .$$

Ce dernier système peut encore s'écrire :

$$\left\{\begin{array}{l}\pi_1\sum_{n=1}^{N}\Pi_{n(T+1)}+\sum_{m=1}^{T}\pi_m\sum_{n=1}^{N}\Pi_{n1}=\sum_{n=1}^{N}\Pi_{n1}\\ ...\\ \pi_k\sum_{n=1}^{N}\Pi_{n(T+1)}+\sum_{m=1}^{T}\pi_m\sum_{n=1}^{N}\Pi_{nk}=\sum_{n=1}^{N}\Pi_{nk}\\ ...\\ \pi_T\sum_{n=1}^{N}\Pi_{n(T+1)}+\sum_{m=1}^{T}\pi_m\sum_{n=1}^{N}\Pi_{nT}=\sum_{n=1}^{N}\Pi_{nT}\end{array}\right. .$$

Nous utilisons le fait que la somme des proportions fasse 1 pour écrire :

$$\begin{cases} \pi_1 = \pi_{T+1} \dfrac{\sum_{n=1}^{N} \Pi_{n1}}{\sum_{n=1}^{N} \Pi_{n(T+1)}} \\ \quad ... \\ \pi_k = \pi_{T+1} \dfrac{\sum_{n=1}^{N} \Pi_{nk}}{\sum_{n=1}^{N} \Pi_{n(T+1)}} \\ \quad ... \\ \pi_T = \pi_{T+1} \dfrac{\sum_{n=1}^{N} \Pi_{nT}}{\sum_{n=1}^{N} \Pi_{n(T+1)}} \end{cases} . \tag{4.8}$$

Si nous sommons toutes les équations nous obtenons :

$$\sum_{k=1}^{T} \pi_k = \pi_{T+1} \frac{\sum_{k=1}^{T} \sum_{n=1}^{N} \Pi_{nk}}{\sum_{n=1}^{N} \Pi_{n(T+1)}}.$$

En se rappelant à nouveau que la somme des proportions π_k pour $k = 1, ..., T+1$ vaut 1 nous obtenons :

$$\pi_{T+1} \frac{\sum_{n=1}^{N} \sum_{k=1}^{T+1} \Pi_{nk}}{\sum_{n=1}^{N} \Pi_{n(T+1)}} = 1.$$

Or, $\sum_{k=1}^{T+1} \Pi_{nk} = 1$, d'où :

$$\pi_{T+1} = \frac{1}{N} \sum_{n=1}^{N} \Pi_{n(T+1)}.$$

D'où la mise à jour pour π_{T+1}. Nous en déduisons facilement la mise à jour pour les autres éléments grâce à (4.8) :

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \Pi_{nk} \quad k = 1, ..., T+1$$

Nous remarquons que cette mise à jour ne dépend pas des densités choisies pour le mélange.

Initialisation de l'algorithme

L'algorithme a besoin d'être initialisé. L'algorithme EM, comme la plupart des méthodes de maximisation numérique ne permet que de trouver un maximum local. C'est pourquoi, d'habitude, une grille de valeurs initiales est proposée, et nous choisissons celles qui permettent de maximiser le plus la

vraisemblance finale. Mais dans notre cas, au vue du nombre de paramètres et de la quantité de données, une telle approche ne serait pas réalisable sans des temps de calculs exagérément longs. C'est pourquoi, nous décidons d'initialiser les paramètres selon les *a priori* biologiques que nous connaissons. Nous initialisons $\lambda_{m_{t_i}^+}$ en calculant le maximum de vraisemblance sur toutes les données pour un temps donné, en choisissant des gènes dont les connaissances biologiques permettent de supposer qu'ils appartiennent au cluster concerné :

$$\lambda_{m_{t_i}^+} = \frac{NP}{\sum_{p=1}^{P}\sum_{n=1}^{N} \Pi_{nm}|x_{npt_i}|}.$$

Les λ_{m*} sont choisis proportionnellement plus petits (environ $0.01\lambda_{m_{t_i}^+}$) parce que le paramètre d'une loi exponentielle est l'inverse de sa moyenne. Par conséquent, des valeurs plus petites de λ permettent de représenter des pics d'une plus grande amplitude. Ensuite, $\lambda_{t_{i^-}}$ est supposé égal à λ_m pour permettre de grandes variations négatives. Les proportions sont supposées égales ; nous les initialisons à $1/(T+1)$.

4.1.2 Modèle linéaire pour l'inférence du réseau

Pour simplifier les notations, nous supposerons ici que $T = 4$. La généralisation se fait facilement.

Suite à l'étape de clustering, nous avons pu sélectionner N_{clust} gènes et attribuer à chacun d'eux son cluster.

Dans l'idéal, nous aimerions poser un modèle de la forme :

$$\boldsymbol{x}_{jp.} = \sum_{n=1,n\neq j}^{N_{\text{clust}}} \mathcal{F}_{ij}(\boldsymbol{x}_{np.}) + \boldsymbol{\eta}_j$$

où nous avons posé :

$$\boldsymbol{x}_{jp.} = \begin{bmatrix} x_{jpt_1} \\ x_{jpt_2} \\ x_{jpt_3} \\ x_{jpt_4} \end{bmatrix} \quad \text{et} \quad \boldsymbol{\eta}_j = \begin{bmatrix} \eta_{jt_1} \\ \eta_{jt_3} \\ \eta_{jt_2} \\ \eta_{jt_4} \end{bmatrix}$$

où $\boldsymbol{\eta}_j$ est un bruit blanc et $\mathcal{F}_{ij}(\bullet)$ une fonction qui modélise l'influence du gène i sur le gène j. Si aucune hypothèse n'est faite sur le modèle, ce dernier ne sera pas identifiable, comme nous l'avons déjà souligné plus haut. C'est pourquoi nous allons supposer le modèle linéaire, et nous allons décomposer $\mathcal{F}_{ij}(\bullet)$ en deux parties de la manière suivante :

$$\boldsymbol{x}_{jp.} = \sum_{i=1}^{N_{clust}} \omega_{ij} F_{m(i)m(j)}(\boldsymbol{x}_{ip.}) + \tilde{\boldsymbol{\eta}}_j \tag{4.9}$$

où ω_{ij} représente la puissance du lien du gène i sur le gène j. Nous supposons que $\omega_{ij} \geqslant 0$, $m(.)$ est la fonction qui à un gène associe son cluster, la matrice $F_{m(i)m(j)}$ représente la manière dont le gène i agit sur le gène j.

Cette matrice peut avoir des coefficients négatifs ; de par son indexation, nous remarquons que nous supposons que l'action d'un gène sur un autre ne dépend pas intrinsèquement des gènes concernés, mais de leur cluster. Nous cherchons alors à minimiser :

$$\mathbf{\Omega}^*, \{F\}^* = \underset{\mathbf{\Omega}, \{F\}}{\arg\min} \sum_{j=1}^{N_{clust}} \sum_{p=1}^{P} \left\| \boldsymbol{x}_{jp.} - \sum_{i=1}^{N_{clust}} \omega_{ij} F_{m(i)m(j)}(\boldsymbol{x}_{ip.}) \right\|_2$$

en posant $\mathbf{\Omega}$ la matrice des ω_{ij} pour $(i,j) \in [\![1, N_{clust}]\!]^2$. Nous rajoutons les contraintes :

$$\sum_{i=1}^{N_{clust}} \omega_{ij} \leqslant \lambda, \ \forall j \in [\![1, N_{clust}]\!], \tag{4.10}$$

avec λ un paramètre de parcimonie choisi par l'utilisateur (ou déterminé par validation croisée).

Nous normalisons ensuite chaque colonne de $\mathbf{\Omega}$ de telle sorte que la somme de chacune d'entre elle fasse 0 (s'il n'y a pas de gène régulateur pour le gène considéré) ou 1.

La deuxième contrainte est une contrainte $\mathcal{L}_1$ de type Lasso. Cette contrainte permet d'obtenir une estimation parcimonieuse. Le vecteur inféré contiendra d'autant plus de 0, que l'inégalité sera sévère. L'estimation simultanée de la matrice $\mathbf{\Omega}$ et des matrices F n'est pas convexe ; c'est pourquoi nous avons recours à une estimation par procédure ascendante. En effet, si la matrice $\mathbf{\Omega}$ est connue, la méthode des moindres carrés permet de trouver les matrices F. Et inversement, si les matrices F sont connues, un algorithme quadratique permettra d'estimer $\mathbf{\Omega}$. Cette méthode assure de trouver un minimum local.

Nous imposons une dernière contrainte à notre modèle. En effet, nous supposons que les matrices $F_{m(i)m(j)}$ (qui se lit : interaction d'un gène i de cluster $m(i)$ sur un gène j de cluster $m(j)$) sont toutes de la forme :

$$F_{m(i)m(j)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ b & a & 0 & 0 \\ c & b & a & 0 \end{bmatrix} \quad \text{avec } a, b, c \in \mathbb{R}^3.$$

Le fait que cette matrice soit triangulaire inférieure permet d'imposer une contrainte de temporalité. En effet, prenons le cas simplifié où seul le gène i de cluster $m(i)$ agit sur le gène j de cluster $m(j)$ pour un patient p donné ; supposons également que $\omega_{ij} = 1$, nous supposons alors que l'interaction s'écrit :

$$\boldsymbol{x}_{jp.} = F_{m(i)m(j)} \boldsymbol{x}_{ip.}$$

ce que nous réécrivons matriciellement :

$$\begin{bmatrix} x_{jpt_1} \\ x_{jpt_2} \\ x_{jpt_3} \\ x_{jpt_4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ b & a & 0 & 0 \\ c & b & a & 0 \end{bmatrix} \begin{bmatrix} x_{ipt_1} \\ x_{ipt_2} \\ x_{ipt_3} \\ x_{ipt_4} \end{bmatrix}$$
$$= \begin{bmatrix} 0 \\ a \times x_{ipt_1} \\ a \times x_{ipt_2} + b \times x_{ipt_1} \\ a \times x_{ipt_3} + b \times x_{ipt_2} + b \times x_{ipt_1} \end{bmatrix}.$$

Remarque 45 *Le premier temps du vecteur $\boldsymbol{x}_{jp.}$ ne peut pas être déduit du vecteur $\boldsymbol{x}_{ip.}$. En fait, la composante $x_{jpt_{k1}}$ correspondant au temps t_{k1} du vecteur $\boldsymbol{x}_{jp.}$ ne peut être déduite que par les composantes $x_{ipt_{k2}}$ correspondant au temps t_{k2} du vecteur $\boldsymbol{x}_{ip.}$ tels que $t_{k1} > t_{k2}$. Biologiquement, cela signifie que nous interdisons les boucles rétro-actives. D'autre part, nous remarquons aussi que la composante $x_{ipt_{k4}}$ n'est jamais utilisée. Ces deux remarques seront importantes dans la suite de notre analyse.*

Par ailleurs, une dernière contrainte de temporalité est imposée. Nous allons supposer qu'un gène d'un cluster $m(i)$ ne peut agir que sur les gènes des clusters $m(j)$ avec $m(i) > m(j)$. Autrement dit :

$$F_{m(i)m(j)} = 0 \text{ si } m(i) \leqslant m(j).$$

Par conséquent, les seules matrices $F_{m(i)m(j)}$ non nulles sont : $F_{12}, F_{13}, F_{14}, F_{23}, F_{24}$, et F_{34}. En conséquence, nous posons également :

$$\omega_{ij} = 0 \text{ si } m(i) \leqslant m(j).$$

Dans la procédure itérative, c'est-à-dire l'estimation de la matrice $\boldsymbol{\Omega}$ puis l'estimation des matrices $F_{m(i)m(j)}$, l'algorithme est initialisé en supposant :

$$\omega_{ij} = \omega_j \text{ si } m(i) > m(j) \text{ tel que } \sum_{i=1}^{N_{clust}} \omega_{ij} = 1.$$

Encore une fois, les sous-sections suivantes sont les détails calculatoires de la méthode. Même si une partie de ces calculs seront changés dans la version modifiée de la méthode, le lecteur peut passer à la section 4.1.3. directement.

Estimation des matrices $F_{m(i)m(j)}$

Nous donnons ici la méthode proposée dans Kemper (2006). Cette méthode n'a pas été retenue dans notre travail et sera améliorée dans le paragraphe suivant. Nous expliquerons les raisons de notre choix.

Cette méthode suppose que les matrices $F_{m(i)m(j)}$ peuvent être estimées de manière totalement indépendantes. Supposons que nous cherchons à estimer la matrice $F_{k_d k_g}$. Nous notons $N_{clust(g)}$ et $N_{clust(d)}$ le nombre de gène dans les clusters k_g et k_d respectivement. Nous notons également $\boldsymbol{\Gamma}$ l'ensemble

des gènes des clusters k_g et $\boldsymbol{D}$ l'ensemble des gènes du cluster k_d. Puisque nous avons P patients dans chacun des clusters, nous avons $P \times N_{clust(g)} \times N_{clust(d)}$ interactions possibles (nous supposons que les interactions mettent en relation les mêmes patients, voir 4.9) mettant chacune en jeu la matrice $F_{k_g k_d}$. Nous pouvons matriciellement les représenter de la manière suivante. Pour ne pas alourdir les matrices, nous supposerons ici que nous n'avons qu'un patient.

$$\underbrace{\begin{bmatrix} \underbrace{\begin{matrix} g_{11t_1} & \cdots & g_{N_{clust(d)}1t_1} \\ g_{11t_2} & \cdots & g_{N_{clust(d)}1t_2} \\ g_{11t_3} & \cdots & g_{N_{clust(d)}1t_3} \\ g_{11t_4} & \cdots & g_{N_{clust(d)}1t_4} \end{matrix}}_{\boldsymbol{\Gamma}} & \cdots & \underbrace{\begin{matrix} g_{11t_1} & \cdots & g_{N_{clust(g)}1t_1} \\ g_{11t_2} & \cdots & g_{N_{clust(g)}1t_2} \\ g_{11t_3} & \cdots & g_{N_{clust(g)}1t_3} \\ g_{11t_4} & \cdots & g_{N_{clust(g)}1t_4} \end{matrix}}_{\boldsymbol{\Gamma}} \end{bmatrix}}_{N_{clust(d)} \text{ fois}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ b & a & 0 & 0 \\ c & b & a & 0 \end{bmatrix}$$

$$\times \begin{bmatrix} \underbrace{\begin{matrix} d_{11t_1} & \cdots & d_{11t_1} \\ d_{11t_2} & \cdots & d_{11t_2} \\ d_{11t_3} & \cdots & d_{11t_3} \\ d_{11t_4} & \cdots & d_{11t_4} \end{matrix}}_{N_{clust(g)} \text{ fois}} & \cdots & \underbrace{\begin{matrix} d_{N_{clust(d)}1t_1} & \cdots & d_{N_{clust(d)}1t_1} \\ d_{N_{clust(d)}1t_2} & \cdots & d_{N_{clust(d)}1t_2} \\ d_{N_{clust(d)}1t_3} & \cdots & d_{N_{clust(d)}1t_3} \\ d_{N_{clust(d)}1t_4} & \cdots & d_{N_{clust(d)}1t_4} \end{matrix}}_{N_{clust(g)} \text{ fois}} \end{bmatrix} + \eta.$$

Si P patients étaient présents, nous aurions de chaque côté de l'égalité des matrices $1 \times P$ par bloc, chaque bloc représentant un patient donné. Les matrices de données sont donc de taille : $4 \times \left[P \times N_{clust(g)} \times N_{clust(d)}\right]$.

Puisque la matrice $F_{k_d k_g}$ a sa première ligne et sa dernière colonne nulle, nous pouvons simplifier l'écriture de la dernière égalité (voir Remarque 45) :

$$\underbrace{\begin{bmatrix} \underbrace{\begin{matrix} g_{11t_2} & \cdots & g_{N_{clust(d)}1t_2} \\ g_{11t_3} & \cdots & g_{N_{clust(d)}1t_3} \\ g_{11t_4} & \cdots & g_{N_{clust(d)}1t_4} \end{matrix}}_{\boldsymbol{\Gamma}} & \begin{matrix} \cdots \\ \cdots \\ \cdots \\ \cdots \end{matrix} & \underbrace{\begin{matrix} g_{11t_2} & \cdots & g_{N_{clust(g)}1t_2} \\ g_{11t_3} & \cdots & g_{N_{clust(g)}1t_3} \\ g_{11t_4} & \cdots & g_{N_{clust(g)}1t_4} \end{matrix}}_{\boldsymbol{\Gamma}} \end{bmatrix}}_{N_{clust(d)} \text{ fois}} = \begin{bmatrix} a & 0 & 0 \\ b & a & 0 \\ c & b & a \end{bmatrix}$$

$$\times \begin{bmatrix} \underbrace{\begin{matrix} d_{11t_1} & \cdots & d_{11t_1} \\ d_{11t_2} & \cdots & d_{11t_2} \\ d_{11t_3} & \cdots & d_{11t_3} \end{matrix}}_{N_{clust(g)} \text{ fois}} & \begin{matrix} \cdots \\ \cdots \\ \cdots \\ \cdots \end{matrix} & \underbrace{\begin{matrix} d_{N_{clust(d)}1t_1} & \cdots & d_{N_{clust(d)}1t_1} \\ d_{N_{clust(d)}1t_2} & \cdots & d_{N_{clust(d)}1t_2} \\ d_{N_{clust(d)}1t_3} & \cdots & d_{N_{clust(d)}1t_3} \end{matrix}}_{N_{clust(g)} \text{ fois}} \end{bmatrix} + \eta.$$

Nous cherchons alors à avoir une écriture de type "régression linéaire". Pour cela, nous réécrivons le système de la manière suivante :

$$\underbrace{\begin{bmatrix} g_{11t_2} \\ g_{11t_3} \\ g_{11t_4} \\ \vdots \\ g_{N_{clust(d)}1t_2} \\ g_{N_{clust(d)}1t_3} \\ g_{N_{clust(d)}1t_4} \\ \vdots \\ g_{11t_2} \\ g_{11t_3} \\ g_{11t_4} \\ \vdots \\ g_{N_{clust(g)}1t_2} \\ g_{N_{clust(g)}1t_3} \\ g_{N_{clust(g)}1t_4} \end{bmatrix}}_{\tilde{\Gamma}} = \underbrace{\begin{bmatrix} d_{11t_1} & 0 & 0 \\ d_{11t_2} & d_{11t_1} & 0 \\ d_{11t_3} & d_{11t_2} & d_{11t_1} \\ \vdots & \vdots & \vdots \\ d_{11t_1} & 0 & 0 \\ d_{11t_2} & d_{11t_1} & 0 \\ d_{11t_3} & d_{11t_2} & d_{11t_1} \\ \vdots & \vdots & \vdots \\ d_{N_{clust(d)}1t_1} & 0 & 0 \\ d_{N_{clust(d)}1t_2} & d_{N_{clust(d)}1t_1} & 0 \\ d_{N_{clust(d)}1t_3} & d_{N_{clust(d)}1t_2} & d_{N_{clust(d)}1t_1} \\ \vdots & \vdots & \vdots \\ d_{N_{clust(d)}1t_1} & 0 & 0 \\ d_{N_{clust(d)}1t_2} & d_{N_{clust(d)}1t_1} & 0 \\ d_{N_{clust(d)}1t_3} & d_{N_{clust(d)}1t_2} & d_{N_{clust(d)}1t_1} \end{bmatrix}}_{\tilde{D}} \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\beta} + \eta.$$

Si $\tilde{\boldsymbol{D}}$ est de rang maximal, alors $\tilde{\boldsymbol{D}}'\tilde{\boldsymbol{D}}$ est inversible et nous pouvons donner l'estimation des moindres carrés suivante pour $\boldsymbol{\beta}$:

$$\hat{\beta} = \left(\tilde{\boldsymbol{D}}'\tilde{\boldsymbol{D}}\right)^{-1} \tilde{\boldsymbol{D}}'\tilde{\boldsymbol{\Gamma}}$$

Pour prendre en compte les différents poids, il faut extraire de la matrice $\boldsymbol{\Omega}$ une matrice de dimension $N_{clust(g)} \times N_{clust(d)}$ correspond aux gènes concernés. Nous appelons $\tilde{\boldsymbol{W}}$ le vecteur obtenu en empilant les colonnes de la matrice extraite. Nous notons $diag(\tilde{\boldsymbol{W}})$ la matrice diagonale dont les éléments diagonaux sont les éléments de $\tilde{\boldsymbol{W}}$. Nous pouvons ensuite utiliser l'estimateur des moindres carrés pondérés :

$$\hat{\beta} = \left(\tilde{\boldsymbol{D}}'diag(\tilde{\boldsymbol{W}})^{-1}\tilde{\boldsymbol{D}}\right)^{-1} \tilde{\boldsymbol{D}}'diag(\tilde{\boldsymbol{W}})^{-1}\tilde{\boldsymbol{\Gamma}}$$

Ce qui achève la méthode proposée. Néanmoins, cette méthode n'est pas satisfaisante, parce qu'elle suppose que les matrices $F_{m(i)m(j)}$ peuvent être estimées séparément.

Inférence du réseau

L'inférence de la matrice $\boldsymbol{\Omega}$ se fait colonne par colonne, c'est-à-dire que nous allons estimer pour j fixé l'ensemble de ω_{ij} pour $i = 1, ..., N_{clust}$. Nous commençons par rappeler l'équation du modèle (4.9) :

$$\boldsymbol{x}_{jp.} = \sum_{i=1}^{N_{clust}} \omega_{ij} F_{m(i)m(j)}(\boldsymbol{x}_{ip.}) + \tilde{\boldsymbol{\eta}}_j.$$

Dans cette partie nous supposons que les $F_{m(i)m(j)}$ sont connues. Nous cherchons alors à minimiser, pour un j fixé :

$$\underset{\boldsymbol{\omega}_{.j}\in\mathbb{R}^{N_{clust}}}{\arg\min} \left\| \boldsymbol{x}_{j..} - \sum_{i=1}^{N} \omega_{ij} \tilde{F}_{m(i)m(j)}(\boldsymbol{x}_{i..}) \right\|_2^2$$

avec $\tilde{F}_{m(i)m(j)}$ est une matrice diagonale par bloc de dimension P, et chaque élément de la diagonale est la matrice $F_{m(i)m(j)}$. Comme $F_{m(i)m(j)}$ et $\boldsymbol{x}_{i..}$ sont connus nous pouvons poser :

$$\tilde{F}_{m(i)m(j)}(\boldsymbol{x}_{i..}) = \begin{bmatrix} d_{1i} \\ d_{2i} \\ d_{3i} \\ \vdots \\ d_{(4P)i} \end{bmatrix}.$$

Nous devons alors rechercher l'argument minimum suivant :

$$\underset{\boldsymbol{\omega}_{.j}\in\mathbb{R}^{N_{clust}}}{\arg\min} \left\| \begin{bmatrix} x_{j1t_1} \\ x_{j1t_2} \\ x_{j1t_3} \\ \vdots \\ x_{jPt_4} \end{bmatrix} - \sum_{i=1}^{N_{clust}} \begin{bmatrix} d_{1i} \\ d_{2i} \\ d_{3i} \\ \vdots \\ d_{(4P)i} \end{bmatrix} \omega_{ij} \right\|_2^2.$$

Nous réécrivons la somme en tant que produit de matrices :

$$\underset{\boldsymbol{\omega}_{.j}\in\mathbb{R}^{N_{clust}}}{\arg\min} \left\| \underbrace{\begin{bmatrix} x_{j1t_1} \\ x_{j1t_2} \\ \vdots \\ x_{jPt_4} \end{bmatrix}}_{\mathbb{X}} - \underbrace{\begin{bmatrix} d_{11} & \dots & d_{1N_{clust}} \\ d_{21} & \dots & d_{2N_{clust}} \\ \vdots & \ddots & \vdots \\ d_{(4P)1} & \dots & d_{(4P)N_{clust}} \end{bmatrix}}_{\mathbb{D}} \underbrace{\begin{bmatrix} \omega_{1j} \\ \vdots \\ \omega_{Nj} \end{bmatrix}}_{\mathbb{W}} \right\|_2^2.$$

Nous développons pour trouver une forme quadratique :

$$\begin{aligned} &\underset{\mathbb{W}\in\mathbb{R}^{N_{clust}}}{\arg\min} \left[(\mathbb{X} - \mathbb{D}\mathbb{W})'(\mathbb{X} - \mathbb{D}\mathbb{W}) \right] \\ &= \underset{\mathbb{W}\in\mathbb{R}^{N_{clust}}}{\arg\min} \left[\mathbb{X}'\mathbb{X} - (\mathbb{D}\mathbb{W})'\mathbb{X} - \mathbb{X}'(\mathbb{D}\mathbb{W}) + \mathbb{W}'\mathbb{D}'\mathbb{D}\mathbb{W} \right] \\ &= \underset{\mathbb{W}\in\mathbb{R}^{N_{clust}}}{\arg\min} \left[-(\mathbb{A}'\mathbb{X})'\mathbb{W} + \frac{1}{2}\mathbb{W}'\mathbb{D}'\mathbb{D}\mathbb{W} \right]. \end{aligned}$$

4.1.3 Conclusion

Nous estimons donc le modèle d'inférence par une procédure ascendante. Nous estimons d'abord les matrices $F_{m(i)m(j)}$ en supposant la matrice $\boldsymbol{\Omega}$ connue, puis nous estimons la matrice $\boldsymbol{\Omega}$ en supposant les matrices $F_{m(i)m(j)}$ connues. Quand la convergence est atteinte, nous obtenons le réseau de gènes grâce à la matrice $\boldsymbol{\Omega}$. Chaque élément non nul de cette matrice correspond à un arc dans le réseau. Compte tenu du bruit inhérent aux données de type

micro-puces, il peut être judicieux d'élaguer la matrice $\mathbf{\Omega}$ en supprimant les liens en dessous d'un seuil fixé à l'avance.

4.2 Modifications du modèle initial

Dans cette section nous allons expliquer quelles modifications ont été faites au modèle, et nous détaillerons pourquoi nous pensons qu'elles étaient nécessaires.

4.2.1 Etape de clustering

L'étape de clustering a été modifiée pour deux raisons principales. La première est que nous souhaitions intégrer des gènes fortement exprimés quel que soit leur pattern d'expression. La deuxième raison est que la loi exponentielle utilisée dans l'équation 4.3 pour le modèle de mélange ne semblait pas être un choix optimal pour modéliser les pics d'activité. En effet, le mode de la loi exponentielle est en 0 alors qu'il devrait être au niveau d'expression moyen ou médian du gène. Le graphique 4.2 explique comment le modèle fonctionne avec la loi exponentielle. La queue de la loi exponentielle étant plus lourde que celle de loi normale, nous préférerons choisir la loi exponentielle pour des observations avec un grand niveau d'expression, tandis que le bruit sera retenu dans la loi normale. Si le modèle proposé initialement parvient à séparer le bruit des observations fortement exprimées, ce dernier ne choisira pas forcément les observations les plus exprimées. Étant donné que nous regardons la probabilité *a posteriori* pour choisir les gènes, il faut à la fois maximiser la densité pour la loi exponentielle et minimiser celle pour la loi normale. Or dans le graphique 4.2, nous voyons que la normale est quasi-nulle à partir de 300. Par conséquent, à partir de cet endroit, il suffit de maximiser la densité exponentielle pour sélectionner un gène. Par conséquent, la probabilité *a a posteriori* sera plus favorable pour un gène qui vaut entre 300 et 400 que pour un gène qui vaut entre 600 et 700. D'autre part, cette modélisation favorise les observations contenant un outlier, comme le montre la Figure 4.3.

Pour pallier ces deux problèmes nous avons changé la manière de faire le clustering :

1. Choisir les N_1 gènes les plus exprimés
2. Parmi ces N_1 gènes garder les N_2 gènes les plus exprimés, et choisir N_3 gènes ayant un pic à un temps donné grâce à un algorithme EM sur les $N_1 - N_2$ gènes restants.

Reste à définir comment ces deux étapes sont réalisées. Pour choisir les gènes les plus exprimés, nous utilisons un package R décrit dans Bhowmick *et al.* (2006a). Un modèle hiérarchique a été choisi dans lequel les observations $y_1, ..., y_n$ sont supposés être des tirages indépendants d'une loi normale Y
:

— $y_1, ..., y_n | \mu, \sigma^2 \sim \mathcal{N}(\mu, \sigma^2)$
— $\mu | \sigma^2 \sim \omega Laplace(0, V\sigma^2) + (1 - \omega)\delta_0$
— $\sigma^2 \sim \mathcal{IG}(\alpha, \gamma)$

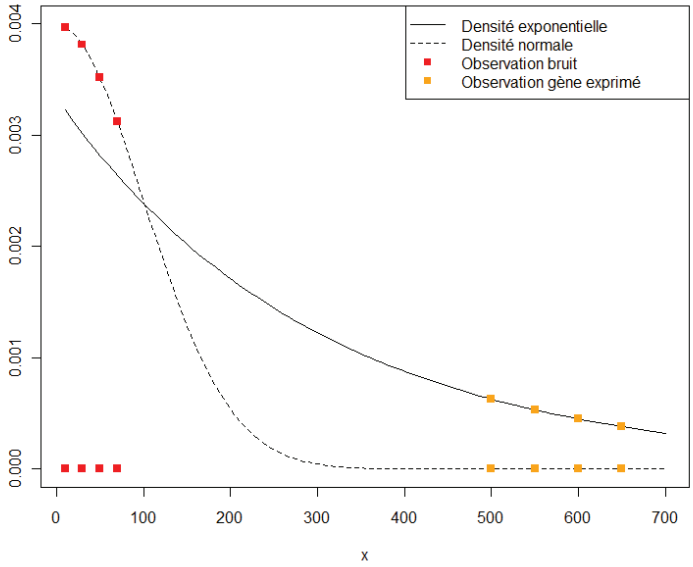


FIGURE 4.2 – *La queue de la loi exponentielle étant plus lourde que celle de loi normale, nous préférerons choisir la loi exponentielle pour des observations avec un grand niveau d'expression, tandis que le bruit sera retenu dans la loi normale.*

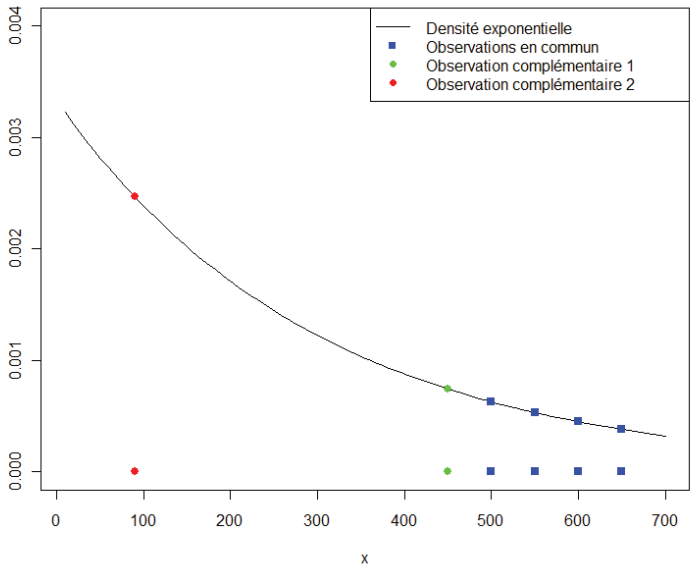


FIGURE 4.3 – *La modélisation exponentielle favorise les observations contenant un outlier*

où $\omega, V, \alpha, \gamma$ sont des paramètres à estimer, et où δ_0 est la mesure de Dirac en 0.

Les détails des calculs sont dans l'article original Bhowmick *et al.* (2006a). Ensuite, pour choisir les gènes ayant un maximum d'expression à un temps donné, nous avons modifié la modélisation en remplaçant la loi exponentielle dans le cas où le gène est censé s'exprimer fortement par une loi de Laplace symétrique, qui peut être considérée comme une "double exponentielle". La densité d'une loi de Laplace est :

$$f(x) = \frac{1}{2b} \exp\left(-\frac{|x-\theta|}{b}\right)$$

avec b un réel positif et θ un réel.

Quelques représentations sont données dans la Figure 4.4.

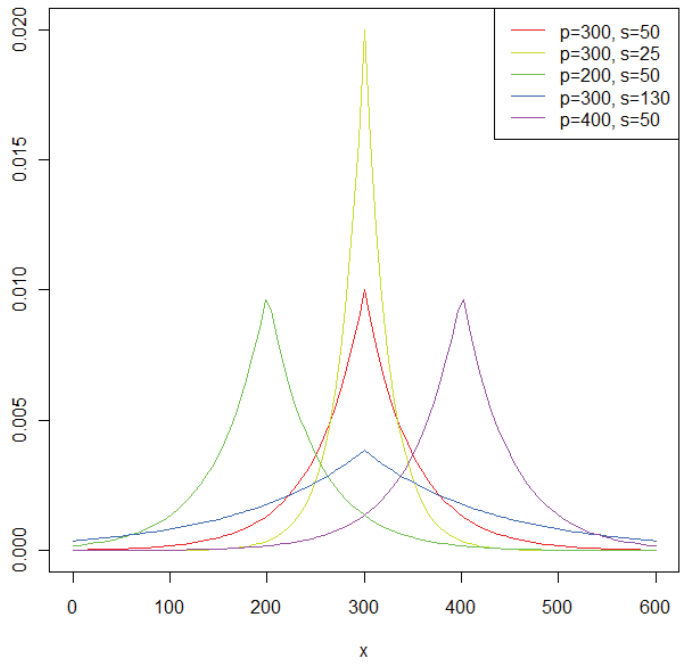


FIGURE 4.4 – *Différentes lois de Laplace, avec p le paramètre de position, et s le paramètre de dispersion.*

La nouvelle modélisation, à comparer avec celle présentée dans l'équation 4.3, est alors, en reprenant les mêmes notations :

$$\begin{cases} \dfrac{1}{2b_m}\exp\left(-\dfrac{|X_{npt_m}-\theta_m|}{b_m}\right) & ; \quad 1\leqslant i\leqslant 4; i=m \\ \dfrac{\lambda_{m_{t_i}^+}}{2}\exp(-\lambda_{m_{t_i}^+}X_{npt_i}) & ; \quad X_{npt_i}>0; 1\leqslant i\leqslant 4; i\neq m \\ \dfrac{\lambda_{t_i^-}}{2}\exp(\lambda_{t_i^-}X_{npt_i}) & ; \quad X_{npt_i}\leqslant 0; 1\leqslant i\leqslant 4; i\neq m \\ \dfrac{1}{2c_{t_i}}\exp\left(-\dfrac{|X_{npt_i}|}{c_{t_i}}\right) & ; \quad 1\leqslant i\leqslant 4; m=5 \end{cases} \tag{4.11}$$

où b_m, c_{t_i}, $\lambda_{t_i^-}$, $\lambda_{m_{t_i}^+}$ sont des réels positifs et les θ_m sont des réels.

L'estimation se fait à l'aide d'un algorithme EM, de la même manière que précédemment. La densité d'une loi de Laplace n'étant pas dérivable sur tout l'ensemble de dérivation (précisément, au niveau du paramètre de position, à cause de la valeur absolue), il n'est pas possible de trouver une formule explicite de mise à jour des paramètres, et c'est pourquoi nous utilisons des algorithmes numériques pour y parvenir.

4.2.2 Estimation des matrices $F_{m(i)m(j)}$

En considérant l'équation (4.9), nous en déduisons que certaines matrices doivent être estimées simultanément :

— F_{12} peut être estimée à part car cette matrice intervient toujours seule
— F_{13} et F_{23} doivent être estimées en même temps
— F_{14}, F_{24} et F_{34} doivent être estimées en même temps.

Estimation de F_{12} Nous repartons de l'équation (4.9). Nous l'écrivons matriciellement, pour tous les j tels que $m(j)=2$ et pour tout $p\in 1,...,P$:

$$\begin{bmatrix} x_{jpt_2} \\ x_{jpt_3} \\ x_{jpt_4} \end{bmatrix} = \sum_{i=1}^{N_{clust}} \omega_{ij} \begin{bmatrix} a_{12} & 0 & 0 \\ b_{12} & a_{12} & 0 \\ c_{12} & b_{12} & a_{12} \end{bmatrix} \begin{bmatrix} x_{ipt_1} \\ x_{ipt_2} \\ x_{ipt_3} \end{bmatrix} + \tilde{\boldsymbol{\eta}}_j$$

Nous réécrivons cette dernière équation sous une forme plus agréable :

$$\begin{aligned} \begin{bmatrix} x_{jpt_2} \\ x_{jpt_3} \\ x_{jpt_4} \end{bmatrix} &= \sum_{i=1}^{N_{clust}} \omega_{ij} \begin{bmatrix} x_{ipt_1} & 0 & 0 \\ x_{ipt_2} & x_{ipt_1} & 0 \\ x_{ipt_3} & x_{ipt_2} & x_{ipt_1} \end{bmatrix} \begin{bmatrix} a_{12} \\ b_{12} \\ c_{12} \end{bmatrix} + \tilde{\boldsymbol{\eta}}_j \\ &= \begin{bmatrix} \sum_{i=1}^{N_{clust}} \omega_{ij}x_{ipt_1} & 0 & 0 \\ \sum_{i=1}^{N_{clust}} \omega_{ij}x_{ipt_2} & \sum_{i=1}^{N_{clust}} \omega_{ij}x_{ipt_1} & 0 \\ \sum_{i=1}^{N_{clust}} \omega_{ij}x_{ipt_3} & \sum_{i=1}^{N_{clust}} \omega_{ij}x_{ipt_2} & \sum_{i=1}^{N_{clust}} \omega_{ij}x_{ipt_1} \end{bmatrix} \begin{bmatrix} a_{12} \\ b_{12} \\ c_{12} \end{bmatrix} + \tilde{\boldsymbol{\eta}}_j \end{aligned}$$

Nous écrivons le système pour tous les patients (de 1 à P) et tous les gènes concernés (nous les renumérotons de 1 à $N_{clust(2)}$) de la manière suivante (nous ne donnons pas tous les détails pour les matrices, l'idée étant simplement que chaque gène apparaisse une et une seule fois) :

$$\begin{bmatrix} x_{11t_2} \\ x_{11t_3} \\ x_{11t_4} \\ \vdots \\ x_{N_{clust(2)}Pt_2} \\ x_{N_{clust(2)}Pt_3} \\ x_{N_{clust(2)}Pt_4} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N_{clust}} \omega_{i1}x_{ipt_1} & 0 & 0 \\ \sum_{i=1}^{N_{clust}} \omega_{i1}x_{ipt_2} & \sum_{i=1}^{N_{clust}} \omega_{i1}x_{ipt_1} & 0 \\ \sum_{i=1}^{N_{clust}} \omega_{i1}x_{ipt_3} & \sum_{i=1}^{N_{clust}} \omega_{i1}x_{ipt_2} & \sum_{i=1}^{N_{clust}} \omega_{i1}x_{ipt_1} \\ \vdots & \vdots & \vdots \\ \sum_{i=1}^{N_{clust}} \omega_{iN_{clust(2)}}x_{iPt_1} & 0 & 0 \\ \sum_{i=1}^{N_{clust}} \omega_{iN_{clust(2)}}x_{iPt_2} & \sum_{i=1}^{N_{clust}} \omega_{iN_{clust(2)}}x_{iPt_1} & 0 \\ \sum_{i=1}^{N_{clust}} \omega_{iN_{clust(2)}}x_{iPt_3} & \sum_{i=1}^{N_{clust}} \omega_{iN_{clust(2)}}x_{iPt_2} & \sum_{i=1}^{N_{clust}} \omega_{iN_{clust(2)}}x_{iPt_1} \end{bmatrix} \begin{bmatrix} a_{12} \\ b_{12} \\ c_{12} \end{bmatrix} + \tilde{\boldsymbol{\eta}}_j \tag{4.12}$$

Nous estimons ensuite les paramètres a_{12}, b_{12}, et c_{12} par moindres carrés ordinaires.

Estimation simultanée de F_{13} et F_{23} Nous distinguons trois ensembles. Nous avons $\boldsymbol{D}$ l'ensemble des gènes du cluster 1, $\boldsymbol{E}$ l'ensemble des gènes du cluster 2, et $\boldsymbol{\Gamma}$ l'ensemble des gènes du cluster 3, et nous notons également d_{npt} le n^{eme} gène pour le patient p au temps t_i pour l'ensemble des gènes du cluster 1, e_{npt} le n^{eme} gène pour le patient p au temps t_i pour l'ensemble des gènes du cluster 2 et g_{npt} le n^{eme} gène pour le patient p au temps t_i pour l'ensemble des gènes du cluster 3. En se souvenant que les gènes du cluster 3 et 4 ne peuvent pas agir sur les gènes du cluster 3 nous pouvons écrire, en partant toujours de l'équation (4.9) :

$$\begin{aligned} \boldsymbol{g}_{jp.} &= \sum_{i=1}^{N_{clust}} \omega_{ij} F_{m(i)m(j)}(\boldsymbol{x}_{ip.}) + \tilde{\boldsymbol{\eta}}_j \\ &= \sum_{i=1}^{N_{clust(1)}} \omega_{ij} F_{23}(\boldsymbol{d}_{ip.}) + \sum_{i=1}^{N_{clust(2)}} \omega_{ij} F_{13}(\boldsymbol{e}_{ip.}) + \tilde{\boldsymbol{\eta}}_j \end{aligned}$$

Comme précédemment, nous allons écrire la version matricielle de cette équation, pour un gène j et un patient p donné du cluster 3 :

$$
\begin{aligned}
\begin{bmatrix} g_{jpt_2} \\ g_{jpt_3} \\ g_{jpt_4} \end{bmatrix} &= \sum_{i=1}^{N_{clust(1)}} \omega_{ij} \begin{bmatrix} d_{ipt_1} & 0 & 0 \\ d_{ipt_2} & d_{ipt_1} & 0 \\ d_{ipt_3} & d_{ipt_2} & d_{ipt_1} \end{bmatrix} \begin{bmatrix} a_{13} \\ b_{13} \\ c_{13} \end{bmatrix} \\
&\quad + \sum_{i=1}^{N_{clust(2)}} \omega_{ij} \begin{bmatrix} e_{ipt_1} & 0 & 0 \\ e_{ipt_2} & e_{ipt_1} & 0 \\ e_{ipt_3} & e_{ipt_2} & e_{ipt_1} \end{bmatrix} \begin{bmatrix} a_{23} \\ b_{23} \\ c_{23} \end{bmatrix} + \tilde{\boldsymbol{\eta}}_j \\
&= \begin{bmatrix} \sum_{i=1}^{N_{clust(1)}} \omega_{ij} d_{ipt_1} & \sum_{i=1}^{N_{clust(2)}} \omega_{ij} e_{ipt_1} & 0 \\ \sum_{i=1}^{N_{clust(1)}} \omega_{ij} d_{ipt_2} & \sum_{i=1}^{N_{clust(2)}} \omega_{ij} e_{ipt_2} & \sum_{i=1}^{N_{clust(1)}} \omega_{ij} d_{ipt_1} \\ \sum_{i=1}^{N_{clust(1)}} \omega_{ij} d_{ipt_3} & \sum_{i=1}^{N_{clust(2)}} \omega_{ij} e_{ipt_3} & \sum_{i=1}^{N_{clust(1)}} \omega_{ij} d_{ipt_2} \end{bmatrix} \\
&\qquad \begin{bmatrix} 0 & 0 & 0 \\ \dots & \sum_{i=1}^{N_{clust(2)}} \omega_{ij} e_{ipt_1} & 0 & 0 \\ \sum_{i=1}^{N_{clust(2)}} \omega_{ij} e_{ipt_2} & \sum_{i=1}^{N_{clust(1)}} \omega_{ij} d_{ipt_1} & \sum_{i=1}^{N_{clust(2)}} \omega_{ij} e_{ipt_1} \end{bmatrix} \begin{bmatrix} a_{13} \\ a_{23} \\ b_{13} \\ b_{23} \\ c_{13} \\ c_{23} \end{bmatrix} + \tilde{\boldsymbol{\eta}}_j
\end{aligned}
$$

Nous faisons ensuite comme dans l'équation (4.12) et nous empilons les différentes équations, et nous terminons par calculer l'estimateur des moindres carrés ordinaires.

Estimation simultanée de F_{14},F_{24} et F_{34} nous suivons la même méthodologie que dans le paragraphe ci-dessus, en distinguant les gènes en fonction de leur cluster.

Conclusion Ceci achève la méthode que nous proposons et qui permet d'estimer simultanément les paramètres qui apparaissent dans une même équation.

4.2.3 Estimation du réseau

La seule modification faite à l'estimation du réseau a été de remplacer la contrainte $\sum_{i=1}^{N_{clust}} \omega_{ij} \leqslant 1$ dans l'équation 4.10 par $\sum_{i=1}^{N_{clust}} \omega_{ij} \leqslant \nu$ où ν est un réel strictement positif à estimer. Cette estimation est faite par validation croisée "leave-one-out" sur l'ensemble des patients disponibles.

Le critère de distance utilisé pour la régression ressemble au R^2 utilisé dans la régression linéaire. Soit $\boldsymbol{X}_{obs}$ la matrice observée, $\boldsymbol{X}_{inf}$ la matrice inférée, et $\mu_{\boldsymbol{X}}$ la moyenne de $\boldsymbol{X}_{obs}$, le critère de distance est alors :

$$d(\boldsymbol{X}_{obs}, \boldsymbol{X}_{inf}) = \frac{\|\boldsymbol{X}_{obs} - \boldsymbol{X}_{inf}\|_2^2}{\|\boldsymbol{X}_{obs} - \mu_{\boldsymbol{X}}\|_2^2}$$

Ce critère a été retenu parce qu'il permet de prendre en compte la variabilité intrinsèque des données.

Rendre le paramètre ν variable semblait essentiel parce qu'il permet à la fois de prendre en compte différents types de structure de réseau de gènes, et permet ensuite de contrôler la précision du résultat en fonction de la précision des données.

4.3 Comparaison de notre méthode d'inférence de réseau

Il existe beaucoup de méthodes d'inférence en réseaux (voir le Chapitre 1 pour une revue). Cependant, aucune de ces méthodes ne traite de manière spécifique les réseaux tels que nous les conceptualisons dans cette thèse, c'est-à-dire en cascade. À partir de ce point, deux questions se posent donc :

— quelles différences observe-t-on sur notre jeu de données entre la méthode que nous proposons et les méthodes dans la littérature ?
— quelle est la performance - en termes de spécificité et de sensibilité - de notre méthode comparée *in silico* avec les méthodes trouvées dans la littérature ?

Ces deux questions ouvrent tout naturellement les deux prochaines sous parties de notre chapitre. Avant cela, nous allons brièvement présenter les méthodes issues de la littérature auxquelles nous nous sommes comparés.

4.3.1 Méthodes pour la comparaison

Nous avons sélectionné quatre méthodes issues de la littérature. Nous avons choisi ces méthodes selon différents critères, parmi lesquels :

— la temporalité : nous disposons de données mesurées au cours du temps, il faut donc que la méthode sélectionnée permette de prendre en compte la temporalité
— la capacité à traiter de grands jeux de données : le réseau que nous voulons inférer est composé de plusieurs centaines de gènes.

Cela nous a conduit à comparer notre méthode avec les méthodes suivantes :

— TD-ARACNE (Zoppoli *et al.* 2010) : cette méthode est basée sur la détection du décalage temporel puis par l'utilisation de l'information mutuelle de la même manière que dans la méthode ARACNE (Margolin *et al.* 2006b)
— Genenet (Schafer et Strimmer 2005) : il s'agit d'un exemple de méthode basée sur les modèles graphiques gaussien qui prend en compte la corrélation partielle
— Morissey (Morrissey *et al.* 2011) : il s'agit d'un exemple de modèle baysésien dynamique, tel que nous les avons définis dans le Chapitre 1
— GeneReg (Huang *et al.* 2010) : il s'agit d'un modèle d'inférence de réseaux basé sur des régressions ; une interpolation par splines des

points de mesure est effectuée pour augmenter artificiellement le nombre de ces points.

—

D'autre part, il faut noter que la méthode de Morissey ne figure pas dans nos résultats car le temps de calcul s'est révélé rédhibitoire (plus d'un mois pour une seule inférence).

4.3.2 Comparaison sur notre jeu de données

L'analyse du jeu de données ainsi que ces implications biologiques sont discutées dans le chapitre suivant. Ici, nous nous permettons simplement d'analyser plus en détail les performances comparées de notre algorithme de reconstruction de réseau.

Une sélection des gènes différentiellement exprimés dans le cas des patients ayant la version agressive de la maladie nous a permis de sélectionner 500 gènes, et c'est sur cette sélection que nous allons comparer les différents algorithmes.

Tout d'abord notons que toutes les méthodes sus-citées intègrent des contraintes de parcimonie : inégalité relative à l'information mutuelle pour TD-ARACNE ou pénalisation pour GeneReg. Il est donc intéressant dans un premier temps de comparer les nombres de liens retenus. Notre méthode ainsi que Genereg aboutissent à un nombre relativement faible de liens dans le réseaux (respectivement 1528 et 1567), GeneNet se situe à un niveau intermédiaire (2241 liens) tandis que TD-ARACNE aboutit à un nombre important de liens dans le réseau (5236 liens).

Ce nombre de liens qui diffère n'est pas surprenant, puisque l'on considère des méthodologies très différentes qui ont chacune leur propre niveau de sensibilité. Ce qui, en revanche, peut surprendre le lecteur est que le pourcentage de liens communs entre les différentes méthodes n'est que de l'ordre de 5% et ce, quel que soit le couple de méthodes choisi. Mais cela s'explique tant par les différences méthodologiques entre les algorithmes, que par le nombre très élevé de liens possibles (250 000) que par le bruit inhérent aux expériences de microarrays que, finalement, par la corrélation linéaire forte présente dans ce jeu de données (voir Chapitre 2). De plus, la concordance des méthode d'inférence de réseaux de gènes est un phénomène bien connu et étudié (Marbach *et al.* 2012).

Cependant, si les différentes méthodes ne détectent pas les mêmes liens, les gènes influents du réseau semblent être partagés. Ainsi, le gène EGR1 régule au moins dix gènes dans les réseaux de régulations géniques obtenus par les quatre méthodes et le gène DUSP1 est un régulateur important pour au moins trois des quatre méthodes (DUSP1 n'est pas un régulateur dans le réseau obtenu par la méthode TD-ARACNE).

4.3.3 Comparaison *in silico*

L'intérêt des comparaisons faites sur des jeux de données simulés est le fait de connaître par avance la topologie du réseau ; c'est pourquoi cette

étape nous semblait être d'une importance capitale. Pour simuler un jeu d'expression de gènes, il est nécessaire de réunir les deux éléments suivants :

— une topologie de réseau
— une simulation dynamique des expressions de gènes basée sur la susmentionnée topologie de réseau

Comme nous l'avons déjà dit, notre modélisation de réseau de gènes est basée sur l'idée de réseau en cascade. C'est pourquoi, dans le cadre de ces simulations, nous avons voulu tester et comparer notre méthode vis à vis d'une topologie classique de réseau invariant d'échelle, et une topologie de réseau invariant d'échelle sous la contrainte temporelle de la cascade. Pour la topologie classique invariante d'échelle, nous avons utilisé le logiciel NEMO (Long et Roth 2008). Pour la topologie de type "Cascade" nous avons utilisé le principe d'attachement préférentiel (Barabási et Albert 1999) que nous avons modifié en intégrant simplement la contrainte de temporalité.

Une fois la topologie du réseau de gènes établie, il faut simuler les expressions de gènes. Pour simuler les expressions de gènes nous nous sommes servis d'un modèle linéaire où l'expression d'un gène au temps t dépend des expressions des ses régulateurs au temps $t-1$. Afin d'obtenir une simulation réaliste, nous avons utilisé la transformation non linéaire suivante :

$$f(x) = \frac{40 * \exp x/3.5}{30 + \exp x/3.5}.$$

Les paramètres de cette fonction ont été choisis de manière arbitraire, tout en veillant à obtenir une fonction avec suffisamment d'amplitude.

Nous avons alors appliqué les quatre algorithmes de reconstruction de réseaux de gènes et nous avons calculé les trois indicateurs suivants :

— sensibilité : VP/(VP+FN)
— ppv : VP/(VP+FP)
— Fscore : 2*sensibilitié*ppv / (sensibilité + ppv),

où nous avons noté VP : vrais positifs, FN : faux négatifs et FP : faux positifs.

Les résultats, présentés en détail dans le chapitre suivant, montre que notre méthode a des performance équivalente aux autres méthodes lorsque la topologie du réseau est de type classique invariant d'échelle, mais qu'elle est largement meilleure tant en terme de sensibilité que de PPV (et donc de Fscore) que toutes les autres méthodes dans le cadre d'une topologie de type "cascade".

4.4 Conclusion

Nous avons présenté dans ce chapitre les outils méthodologiques pour l'inférence des réseaux en cascade. Dans le chapitre suivant, nous allons mette en œuvre cette méthodologie en l'appliquant au jeu de données présenté dans le Chapitre 3. Nous y discuterons les résultats obtenus, en particulier d'un point de vue biologique où nous regarderons quelles sont les fonctions des gènes sélectionnés, quelles sont les fonctions des gènes dits hubs... D'autre part, notre méthode sera également validée d'un point de

vue biologique. En effet, une expérience d'intervention consistant à inhiber l'expression du gène DUSP1 sera menée. Nous montrons que, jusqu'à une certaine limite, notre modèle est capable de prédire l'expression des autres gènes suite à l'inhibition de DUSP1. Ceci est une étape absolument importante puisque après avoir révélé la structure du système biologique que nous considérons, nous faisons, par ce succès en prédiction, un pas important vers la contrôlabilité de ce système.

5

Reverse-engineering the genetic circuitry of a cancer cell with predicted intervention in chronic lymphocytic leukemia

Cette article a été publié dans la revue PNAS Vallat *et al.* (2013). Dans cette article est introduit le concept de réseau de cascade, ainsi qu'une modélisation statistique adaptée. Nous proposons donc une méthode d'inférence de réseau générale, particulièrement bien adaptée dans le cadre de réseaux en cascade. De plus, nous prouvons qu'il est possible de prédire l'expression des gènes après une expérience d'intervention (ici, inhibition du gène DUSP1). Des informations supplémentaires sont disponibles dans l'Annexe A.

5.1 Abstract

Cellular behavior is sustained by genetic programs that are progressively disrupted in pathological conditions, notably cancer. High-throughput gene expression profiling has been used to infer statistical models describing these cellular programs and development is now needed to guide orientated modulation of these systems. Here we develop a regression-based model to reverse-engineer a temporal genetic program, based on relevant patterns of gene expression after cell stimulation. This method integrates the temporal dimension of biological rewiring of genetic programs and enables the prediction of the effect of targeted gene disruption at system level. We tested the performance accuracy of this model on synthetic data before reverse-engineering the response of primary cancer cells to a proliferative (pro-tumorigenic) stimulation in a multistate leukemia biological model i.e. that of chronic lymphocytic leukemia. To validate the ability of our method

to predict the effects of gene modulation on the global program, we performed an intervention experiment on a targeted gene. Comparison of the predicted and observed gene expression changes demonstrate the possibility of predicting the effects of a perturbation in a gene regulatory network, a first step toward an orientated intervention in a cancer cell genetic program.

5.2 Introduction

Cellular behavior is conditioned mostly by functional genetic programs in response to various environmental signals, as initially shown in simple organisms (Lee *et al.* 2002, Luscombe *et al.* 2004). External stimuli activate cellular surface receptors which trigger multiple signaling cascades in cells. The ultimate targets of these cascades are transcription factors that initiate sequential transcriptional activations with high temporal coordination. The first activated genes, at early time points, after cell stimulation, essentially have a fast and transient expression. Their gene products activate expression of various target genes downstream of transcriptional regulation cascades. These latter genes have longer lasting expression and their products sustain the adapted cellular response to initial environmental stimulation (Yosef et Regev 2011). These functional molecular networks are disrupted in various pathologies, e.g. cancer, where genetic aberrations lead to tumoral cellular programs. Since the first application of high-throughput technologies for measuring gene expression, a number of methods have been proposed to reverse-engineer gene regulatory networks ; considered to be the underlying structure of these genetic programs (Barabasi et Oltvai 2004). These different methods were developed to infer gene potential interactions and to describe these networks at system level (Kitano 2002c). The next important goal was to develop statistical tools allowing to control these systems (Liu *et al.* 2011). One of the key challenges is to find out critical genes whose perturbed expression drive these pathological genetic programs toward targeted states. We propose here a predictive method that is able to predict changes in gene expression upon intervention in the network. Predicting the resulting dynamic gene expression after specific targeted gene disruption is a first step toward controllability.

Among statistical approaches developed to reverse-engineer statistical links between genes and to infer underlying gene regulatory programs (Hecker *et al.* 2009) there is as yet no standard method, as each one is based on strong and specific modeling assumptions, indispensable to make the model identifiable (Marbach *et al.* 2010). As we aimed to understand the temporal dynamic of the network, we focused on methods suited for time series gene expression data. These methods can be grouped into three categories : a) information theoretic models which define a proximity measure between genes, b) optimization methods which use a scoring function to choose the best suited network, and c) regression and other systems of equation methods with a prior network structure. Information theoretic models can only be used for descriptive purposes (i.e. no prediction is possible) but are computationally efficient, making them appealing for large data sets. Several proximity criteria may be used, e.g., the partial Pearson correlation coefficient in Graphical Gaussian Models (Schafer et Strimmer 2005) or entropy

in the Time-Delay ARACNE (TD-ARACNE) method (Zoppoli *et al.* 2010). Optimization methods comprise mostly algorithms using discretized gene expression data and are not computationally efficient for large data sets. Equations-based models impose an underlying structure on the gene network (Gardner *et al.* 2003). These last methods were retained in this study because they have led to promising results due to their flexibility (allowing structural prior information to be incorporated in the model), their ability to infer large scale network and their suitability for prediction purposes (Hecker *et al.* 2009).

To develop and test such statistical models, we previously developed a pertinent biological model using human blood cancer cells (Vallat *et al.* 2007). This biological model allowed us to focus on a genetic program which sustains the leukemic process after a cellular stimulation in primary malignant lymphocytes (Stevenson et Caligaris-Cappio 2004, Messmer *et al.* 2004). Furthermore, this model includes various cell states, from healthy (normal) lymphocytes to those implicated in indolent and aggressive chronic lymphocytic leukemia (CLL), allowing us to compare the genetic program of these different cell states which leads in turn to specific proteomic phenotypes (Perrot *et al.* 2011). CLL is defined by a clonal proliferation of B-lymphocytes which accumulate in the blood to form a leukemia that progressively evolves and is currently incurable (Chiorazzi *et al.* 2005). The mechanism of this proliferation is not well understood, but current hypotheses are in favor of a chronic antigenic stimulation of certain lymphocytes as the primary event in tumorogenesis. Indeed stimulation through the B-cell antigen receptor (BCR) is crucial for physiological development and is the basis of immunological response of these cells. However in CLL (as in other leukemias and lymphomas) a sustained and chronic stimulation of unknown origin is thought to chronically stimulate some lymphocytes, progressively leading to a cell transformation and finally - with accumulation of genetic abnormalities - to an autonomous leukemic cell expansion program (Stevenson et Caligaris-Cappio 2004, Chiorazzi *et al.* 2005). Several prognostic subgroups of CLL have been described, encompassing patients with different survival time (Hamblin *et al.* 1999). Gene expression profiles have been assessed in these different leukemic states (Vallat *et al.* 2007, Herishanu *et al.* 2011, Guarini *et al.* 2008) but no comprehensive lymphocyte BCR genetic program has been proposed to date. Inferring a statistical model of the BCR gene program to predict the key genes that need to be ultimately silenced in order to modulate the leukemic genetic program in an oriented way, would enable better drug development in this presently incurable disease. Furthermore, such an approach would be transferable to other cancers and non-malignant complex diseases.

In this study we selected genes using a two-step algorithm which retains genes with high differential expression and genes with specific temporal patterns. We then reverse-engineered the gene regulatory network with a penalized regression-based method. To assess the possibility of controlling such a genetic program, we performed an RNAi knock-down experiment on a targeted gene, predicting the changes in gene expression from wild type to the knock-down cells.

5.3 Results

5.3.1 Gene selection and network reverse-engineering

After cell stimulation, a specific genetic program is initiated by the concerted expression of a limited number of genes. When captured through temporal genome-wide transcriptional data, the expression of these genes of interest needs to be separated from the residual cellular transcription. So, at each time point after stimulation, we studied gene expression both in stimulated cells and in control (unstimulated) cells. Given that several temporal gene expression profiles have revealed complex gene expression after cellular stimulation Yosef et Regev (2011), Hao et Baltimore (2009), we considered that genes with both high expression level and those with a specific expression pattern (regardless of their expression level) are relevant in the program Di Camillo *et al.* (2012). Gene selection methods based upon selection of highly differentially expressed genes are widely used. In this study, highly expressed genes are selected using common statistical methods (Bhowmick *et al.* 2006b) and genes with specific temporal expression patterns are selected with a specific mixture model, which is also used to group genes into time clusters.

After selecting genes that are likely to participate in the genetic program, we specified a regression-based model to reverse-engineer the gene network. To make the model identifiable and interpretable, some biological constraints were assumed. First, we use the time clusters induced by the mixture model to ensure the temporal causality (i.e. if gene n1 is in the time cluster c1 and gene n2 is in the time cluster c2, gene n1 may interact with gene n2 if and only if c2 $>$ c1). More importantly, topological changes have been observed in gene regulatory networks across time (Luscombe *et al.* 2004, Califano 2011). This property implies a variance in the links between genes through time, allowing specific links activation at specific periods of time after cell stimulation. There are only a few methods allowing such a temporal rewiring. Assuming the widespread hypothesis of sparsity of large networks Barabasi et Oltvai (2004), we put a Lasso penalty on the model (Christley *et al.* 2009). As a result, we propose a scalable time rewiring reverse-engineering method, well-suited for large data sets (see Materials and Methods).

5.3.2 Application to synthetic data

In order to test our model for inference purposes and determine how accurate the inferred network is, as compared to the real network, we used synthetic simulated data where the true network is perfectly known. We compared two network topologies for our simulations : W1, which has a scale-free topology ; generated with the RANGE algorithm (Long et Roth 2008), and W2 which has a temporal cascade topology closer to a biological model of transcriptional activation after transient cell stimulation (Yosef et Regev 2011, Alon 2007). These networks are composed of 500 and 300 genes respectively, both with four time points (the number of genes and time points was chosen with the perspective of studying our biological data set). The gene expression was simulated using a non linear logistic function (Weaver *et al.* 1999). We then calculated three usual indicators (Zoppoli

et al. 2010, Bansal *et al.* 2007) : sensitivity, which describes the proportion of detected links among those that are in the real network, predicted positive value (PPV), which describes the proportion of inferred links that are in the real network, and the F-score (Van Rijsbergen 1979) which combines both and therefore is a convenient way to assess the global performance of an inference method.

With the stable state synthetic network generated with RANGE algorithm, our method achieves an F-score = 0.011 (p=0.001), which considerably increases with a temporal cascade network reaching an F-score = 0.159 (p<0.001).

To go further in this evaluation with synthetic data, we sought to compare these performances with those of actual benchmarked algorithms encompassing several mathematical approaches : TD-ARACNE, an information theoretic method Zoppoli *et al.* (2010) ; GeneNet, a Graphical Gaussian Method (Schafer et Strimmer 2005) ; GeneReg, a regression based method (Huang *et al.* 2010) ; and a dynamic Bayesian network method (DBN) by Morrissey et al. (Morrissey *et al.* 2011) (settings and short descriptions of these methods are presented in Tables A.1 and A.2).

Despite the performances of the DBN method (Morrissey *et al.* 2011), its low computational efficiency did not allow to reach any results with such synthetic data size. GeneReg (Huang *et al.* 2010) did not give any significant result for either of the performance indicators. All three remaining methods (TD-ARACNE, GenNet and our method) performed equally on the Range network, with an F-score of 0.01 +/- 0.001. One remarks that a slight change in F-score (for example from 0.011 for our method network to 0.009 for GeneNet) induces an important change in terms of p-value (respectively 0.001 to 0.032). This seems to reveal how difficult it is to reverse-engineer such a 500-nodes network. When using cascade topology network, performances of all methods (TD-ARACNE, GenNet and our method) increased. Nevertheless in this case, our method has much better results with an F-score=0.16, whereas other methods have an F-score inferior to 0.044. The two proposed network topologies are reliable and the true targeted network may be half way between the two. Since our method outperforms the others in both networks, our proposed algorithm appears to be effective in all cases. Detailed results of algorithms comparisons are presented in Table 5.1.

	Our method			
	Sensitivity	PPV	F-score	p-value
Range network topology	0.021(*)	0.007(*)	0.011(*)	0.001
Temporal cascade topology	0.276(*)	0.111(*)	0.159(*)	<0.001

	TD-ARACNEZoppoli *et al.* (2010)			
	Sensitivity	PPV	F-score	p-value
Range network topology	0.062(*)	0.005(*)	0.010(*)	0.006
Temporal cascade topology	0.023(*)	0.040(*)	0.029(*)	<0.001

	GeneNet Schafer et Strimmer (2005)			
	Sensitivity	PPV	F-score	p-value
Range network topology	0.031(*)	0.005(*)	0.009(*)	0.032
Temporal cascade topology	0.071(*)	0.038(*)	0.044(*)	<0.001

	GeneReg Sherman *et al.* (2009)			
	Sensitivity	PPV	F-score	p-value
Range network topology	0.252	0.003	0.007	0.476
Temporal cascade topology	0.655	0.010	0.019	0.895

* : significant at 0.05 ; explicit p-values are
for the F-score.

TABLE 5.1 – *Modelling performances comparisons on synthetic data with other benchmarked methods.*

5.3.3 Application to the CLL data set

We used gene expression data generated and previously reported (Vallat *et al.* 2007). Briefly, three different cell populations (6 healthy B-lymphocytes, 6 leukemic CLL B-lymphocyte of indolent form and 5 leukemic CLL B-lymphocyte of aggressive form) were stimulated in vitro with an anti-IgM antibody, activating the B-cell receptor (BCR). We analyzed the gene expression at four time points (two early time points at 60 and 90 minutes, one intermediary time point at 210 minutes and one late time point at 390 minutes). For each time point, gene expression measurement was performed both in stimulated cells and in control unstimulated cells ; then data were pre-processed using the dChip software (Li et Wong 2001).

The gene selection process retained genes that were highly differentially expressed ($\sim$ 40%) and genes with specific temporal patterns ($\sim$ 60%). Among the 54,675 probe sets, 960 were retained for further analysis. Around 500 genes are retained by cell category ; the distribution of these genes wi-

thin the three cell groups is shown in a Venn diagram (Figure A.1). A core of 183 genes is used by all cell groups. Among these, 118 correspond to unique genes. The exploration of their biological function through the NIH DAVID database (Sherman *et al.* 2009) allows evaluation of the significance of biological function enrichment of this list of genes. The majority of these genes are indeed known to be expressed in response to cellular stimulation (51 genes out of 118, p-value with False Discovery Rate (FDR) correction= 0.0001) and specifically in the gene expression regulation after cell stimulation (44/118, p = 0.0006). Furthermore, the genes shared by the three cell categories are enriched with genes having a transcriptional activity (22/118, p = 0.0003) or a transcriptional regulation activity (26/118, p = 0.0017). As expected, some of these genes are also involved in the BCR signaling regulation through MAP kinase phosphatases (3/118, p = 0.05). Some genes are known to be involved in the biological process of immune regulation (20/118, p = 0.0045) and more specifically in lymphocyte activation (8/118, p = 0.0016). These genes, which are the basis of the response to BCR stimulation within the three cell groups, have labels that are distributed across the four temporal cluster types. Other genes are either shared by two cell groups or are specific to a cell population. More genes (183+86) are shared by the aggressive or indolent leukemic cells than by the healthy cells and the leukemic cells. The differential expression levels of the retained genes as a function of time is shown for a representative patient in Figure 5.1.

The genetic program induced within each cell group is then inferred with a Lasso regression-based method and is represented by a predictive linear model, adjusted independently on each of the three cell groups (see Materials and Methods). Within the model, the expression of one particular gene at a given time point influences the expression of other genes at subsequent time points satisfying the temporal constraint of the gene program. This model defined a network of the probable genetic interactions involved in cell response to antigen stimulation. The inferred network in the three cell categories is shown in Figure 5.2. These models show a scale-free-like structure, where a large fraction (93% in the most aggressive leukemic B-cells) of genes have a small number of outgoing edges (less than 10) and a small fraction of genes, the so-called hub genes, (1% : 7 genes) have a large number of outgoing edges (more than 40). There are two hubs in healthy cells, four in indolent leukemic cells and seven in aggressive leukemic cells. Among these ten hub genes, four are known genes with transcription factor activity (EGR1, EGR3, JUNB, NR4A1), involved in transcriptional activation of the JNK MAP kinase signaling and ERK signaling pathways, downstream of the BCR. Some of these genes are also directly involved in MAP kinase signaling (DUSP1, DUSP2) and in lymphocyte function regulation (CD83). Interestingly, EGR1, which is common to all three cell groups (i.e. it is one of the 183 common genes) appears as a major hub in all three networks. Additionally, the leukemic cells share an important hub gene, DUSP1, as shown in Figures 5.2 a and b. The temporal evolution of the signal is shown in Figure A.2. Genes that are active in the two earlier time points are massively linked whereas genes that are active in the latest time points have much less connections.

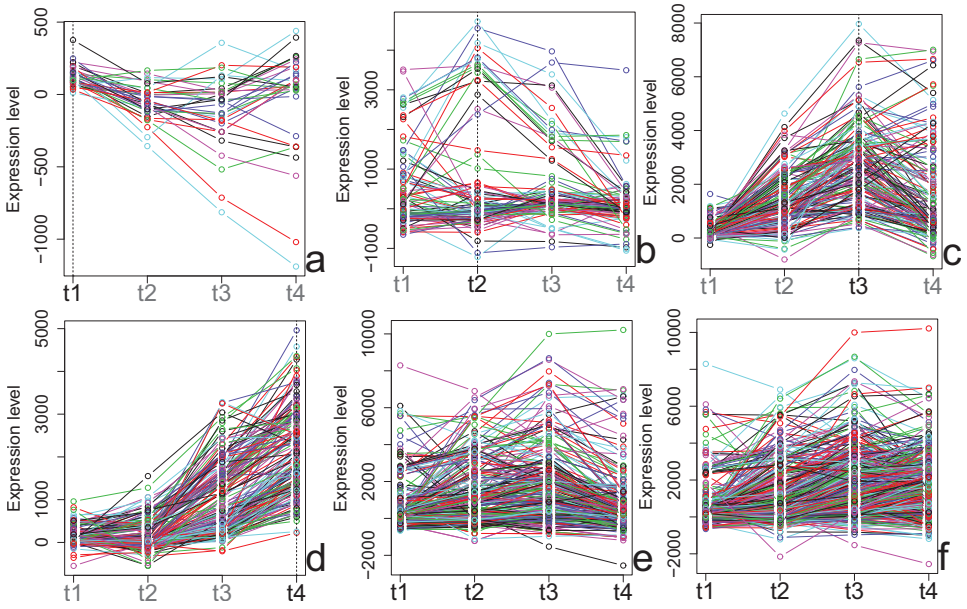


FIGURE 5.1 – *Results of gene selection. Representation of selected genes for a representative patient. Graphs (a) to (d) successively represent genes that have consistent up-regulation at a given time stressed in bold (t_1 to t_4 respectively). Graph (e) shows genes that are highly expressed through all four time points. Graph (f) shows all the retained genes.*

While the structure and parameters of such models provide insight into the nature of a cell gene regulatory network under a given stimulation, the predictive aspect is its main interest. However, the nature of the inferred network is essentially statistical and further experimentation is necessary to distinguish causal from correlated behavior. Perturbation experiments are the usual mechanisms for assessing causal behavior. Consequently, as a feasibility experiment we examined the structure of the inferred network and identified DUSP1 as a candidate gene. DUSP1 is a hub gene in both aggressive and indolent networks (Figure 5.2). It shows up-regulation at the first time point which provides opportunities to measure the effect of perturbing it at later time points on the genes to which it is connected. Furthermore, it has a localized sub-network (Figure 5.2 d) so that effects due to perturbation of DUSP1 can be distinguished from effects following general cell perturbation concomitant to cell transfection. We performed a biological intervention experiment using fresh primary negatively selected B-cells from one aggressive CLL case (see Materials and Methods). We silenced expression of DUSP1 by transfecting DUSP1-specific RNAi and, as a control, transfected cells with a non-targeting RNAi (Figure A.3). We then stimulated the BCR of these cells as previously described (Vallat *et al.* 2007). Whole genome expression profiling was performed at four time points after BCR stimulation, using the same HG-U133+2.0 microarray and pre-processed using dChip (data accessible in GEO database). Gene expression profiles under DUSP1 silencing were then compared to model predictions in which the expression of DUSP1 is set to 0. In this model, the predicted expression is either up-regulated, down-regulated or constant (Figure A.4). For each probe set, prediction is done for the last three time points measurement. Consequently, for each probe set, we can have 0 to 3 correct predictions. Considering our data, where the proportion in the three categories are not equivalent (the

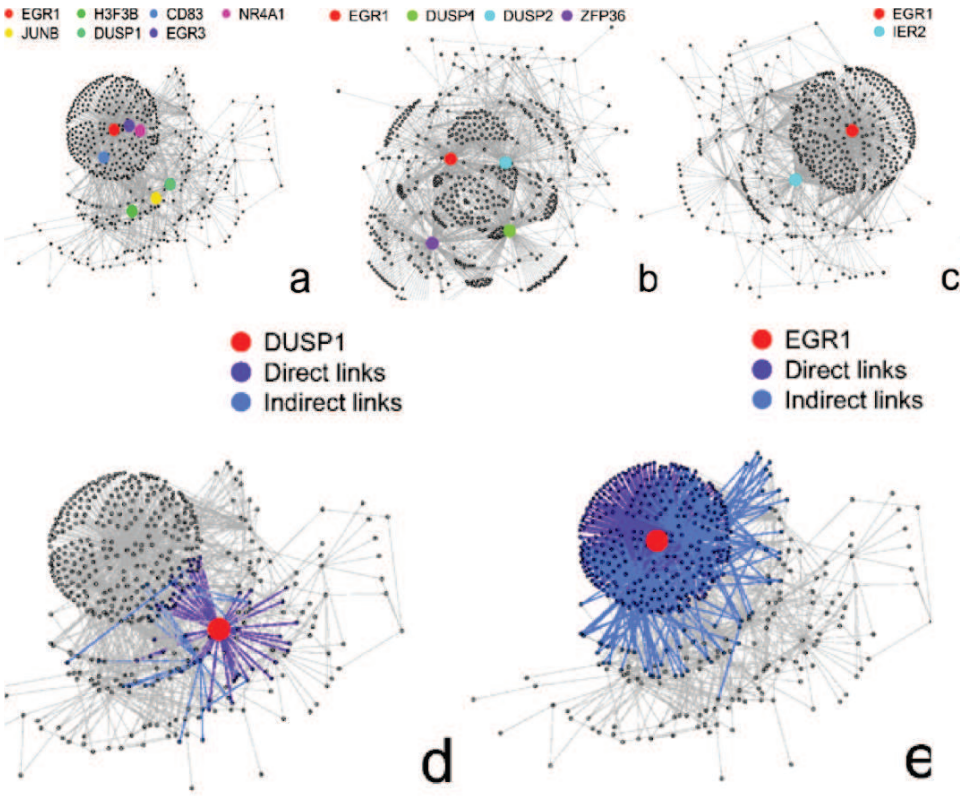


FIGURE 5.2 – *Visualization of inferred networks. The gene regulatory network of the most aggressive leukemic B-cells (a), the indolent leukemic B-cells (b), and healthy B-cells (c) are represented. Nodes represent genes and edges statistical relationships between genes. For each network, hubs are highlighted in color. As the number of hubs decreases between aggressive, indolent and healthy networks, the structure of the network is changed. Bottom graphs represent sub-networks for DUSP1 (d) and EGR1 (e) in the most aggressive leukemic B-cells network. The concerned gene is highlighted in red. Direct links are represented in navy blue and indirect links are represented in pale blue. EGR1 is a gene whose influence is very large, since its subnetwork takes a large part of the complete network. In contrast, DUSP1 has a limited subnetwork. Visualization generated using R and R package Igraph.*

	t_2	p-value	t_3	p-value	t_4	p-value
Linked	62 %	0.004	54%	0.08	43 %	0.70
Not linked	56%	<0.001	59 %	<0.001	40 %	0.97

TABLE 5.2 – *Percentage of correct predictions between observed and inferred network after the silencing of DUSP1.*

number of up-regulated, down-regulated and constant gene expressions are different), the random prediction of one of these three categories is correct with a probability of 45%. However, the observed modulation of expression in this experiment shows 62% correct predictions for genes with a direct link to DUSP1 at t_2 (p-value 0.0041) (Table 5.2). At later time points, the predictive accuracy decreases (t3 : 54%, p-value = 0.08, and t_4 : 43%, p-value = 0.7). At t_4 our predictions are not significantly better than noise. This can be explained by a slow accumulation of the errors, as predictions for time t_4 take into account predictions made at time t_2 and t_3. Although the predictive power of our model decreases in the later time points, results are promising and demonstrate the possibility of an oriented modulation of the gene regulatory network in future work.

5.4 Discussion

We developed a general statistical method for analyzing gene expression as a mean to infer a temporal regulatory network. We first ascertained the performance of this method on synthetic data before analyzing biological data sets. We applied this method to model the response of three different cell groups - healthy B-cells, indolent CLL cells and the most aggressive CLL B-cells - in response to an in vitro stimulation. The results demonstrate different patterns of the genetic program used by each cell group after antigenic stimulation, as shown in the graphical representation of the inferred networks (Figure 5.2). When focusing on the genetic program of the more aggressive leukemic cells, several points of convergence (overlap) are found in the networks inferred by our method and by other benchmark methods (Table A.3). Considering specific topologies of these networks, EGR1 appears as a hub (regulating here more than ten others genes) for all the methods, whereas DUSP1 only appears as a hub for our method and GeneNet. Still focusing on the more aggressive leukemic cells, we used our in silico model to predict the effects of perturbing the genetic program of these cells. This prediction ability imposes specific constraints on model inference (Figure A.5). Obtaining multiple points of measurements via microarray experiments also poses a great challenge when analyzing human cells. Thus, the study deals with a relatively small number of subjects, time points, and points of measurement, including a total of 152 microarrays. The inference method, as a result, explicitly imposes sparseness in the inferred network. The preliminary results suggest the feasibility of such an approach for oriented genetic program modulation. Furthermore, 20% (183 of 960) of the probe sets are shared by the three networks within separate analyses. This suggests the need for further study toward an understanding of how such networks are related and how such networks evolve from a healthy state to the more aggressive state and why, as a consequence, genes are specific to one state

(healthy, indolent or aggressive). To solve this issue we may create a network inferred with all the patients irrespective of their category. However, in such a model, an interaction between two genes might depend on both the incoming stimulation and the state of the considered cell. Furthermore, as shown in the perturbation experiments, analysis of the network structure of such statistical models identifies target genes, typically hubs, for modulation. Ultimately, we should target those genes whose expression can be perturbed under the model in a way leading to an oriented modulation of the cancer cell phenotype. For the particular genetic background and cancer stage of each patient, the method could be used to generate personalized models enabling patient-specified modulations of these cancer disrupted cellular programs.

5.5 Materials and methods

Genes are initially under two states : stimulated and unstimulated (control situation). Their differential expression profiles were computed by subtracting unstimulated from stimulated expression levels at each of the measured time points. Furthermore, a data set $\boldsymbol{X}$ containing N genes, P patients within a sub-population, and 4 time points : $t_1, ..., t_4$ was considered. In this study, each subpopulation (healthy, indolent and aggressive) is modelled separately.

5.5.1 Gene selection

Gene selection was done in two steps. First we selected a large number of highly expressed genes based on a Laplace mixture model (step 1) (Bhowmick *et al.* 2006b). We then used a mixture model, estimated by an expectation-maximization (EM) algorithm, to select, among the remaining genes, those with a specific pattern of expression (step 2). In the mixture model, gene expressions were assumed to come from a finite mixture of probability distributions, with each mixture component $m = 1, ..., 5$ corresponding to a different cluster. In our case, clusters $m = 1, ..., 4$ indicate localized up-regulation of a gene at time t_m and cluster $m = 5$ indicates a gene which is not strongly affected by BCR stimulation and is hence excluded from further analysis. While the parametrization across sub-populations is the same, the actual parameters differ. Formally, we assume that we want to maximize the following likelihood function :

$$L(\Phi; \boldsymbol{X}) = \prod_{n=1}^{N} \sum_{m=1}^{5} p(\boldsymbol{X}_{n..}|m, \Theta_m)\pi_m,$$

where :

$$\Phi = (\pi_1, ..., \pi_M, \boldsymbol{\Theta})',$$

and $\sum_{m=1}^{5} \pi_m = 1$, $\pi_m \in (0, 1)$, for all m, $\boldsymbol{\Theta}$ contains all the parameters $\Theta_1, ..., \Theta_5$ assumed to be distinct, and $\boldsymbol{X}_{n..}$ is the vector expression for gene n across all patients and time points.

The mixture proportions for each cluster are π_m. Conditional probability for a given gene $\boldsymbol{X}_{n..}$ in a given cluster is defined as :

$$p(\boldsymbol{X}_{n..}|m,\Theta_m) = \prod_{p=1}^{P}\prod_{i=1}^{4} p(X_{npt_i}|m,\Theta_m).$$

The subscripts on X specify gene n, patient p and time-point t_i. For purposes of categorization only, time points are modelled as independent. Additionally, the model enforces a common labelling of a given gene across all subjects within a subpopulation. Consequently, disease-related genes exhibiting consistent temporal structure across subjects within a given subpopulation will have a sharp posterior probability under the model, while those which respond to BCR stimulation, but which vary in their response within a sub-population will not. Following convergence of EM fitting, each gene for all P patients within a sub-population is assigned to the cluster with maximum a posteriori probability. We developed a simple parameterization of $p(X_{npt_i}|m;\Theta_m)$ to account for the observed predominance of up-regulation at the specified time points in differential expression. Specifically, genes responding to BCR stimulation are fit to the following model for $p(X_{npt_i}|m;\Theta_m)$:

$$\begin{cases} \dfrac{1}{2b_m}\exp\left(-\dfrac{\left|X_{npt_m}-\theta_m\right|}{b_m}\right) & ; \quad 1\leqslant i\leqslant 4; i=m \\ \dfrac{\lambda_{m_{t_i}^+}}{2}\exp(-\lambda_{m_{t_i}^+}X_{npt_i}) & ; \quad X_{npt_i}>0; 1\leqslant i\leqslant 4; i\neq m \\ \dfrac{\lambda_{t_i^-}}{2}\exp(\lambda_{t_i^-}X_{npt_i}) & ; \quad X_{npt_i}\leqslant 0; 1\leqslant i\leqslant 4; i\neq m \\ \dfrac{1}{2c_{t_i}}\exp\left(-\dfrac{\left|X_{npt_i}\right|}{c_{t_i}}\right) & ; \quad 1\leqslant i\leqslant 4; m=5 \end{cases} \tag{5.1}$$

where b_m, c_{t_i}, λ_{t_i}, $\lambda_{m_{t_i}}$ are positive real numbers and θ_m are real number. These parameters are estimated by the EM algorithm. The use of exponential and Laplacian distributions better captures the heavy-tailed behavior observed in responding genes. The statistical significance of the resulting model was computed using a permutation approach (Mielke et Berry 2007, Ernst *et al.* 2005) and significance computed by comparing the loglikelihood score from EM fitting of the original unpermuted data to the distribution of scores obtained using different permutations for each gene within a trial. Moreover, our clusters are validated by an unsupervised clustering method (Figure A.6). The list of selected genes consists in both the highly differentially expressed genes (step 1) and genes with a specific expression pattern (step 2). Let N_{sel} be the length of this list. We eventually attribute a categorical label to each selected gene describing at which time point its expression is the highest. In the following, let m(i) be the categorical label of gene i.

5.5.2 Model inference

After selecting the genes as described above, we define a linear predictive model :

$$\boldsymbol{x}_{jp.} = \sum_{i=1}^{N_{sel}} F_{m(i)m(j)}\omega_{ij}\boldsymbol{x}_{ip.} + \boldsymbol{\eta}_{j.} \tag{5.2}$$

where :

$$\boldsymbol{x}_{jp.} = (x_{jpt_1}, x_{jpt_2}, x_{jpt_3}, x_{jpt_4})',$$

and :

$$\boldsymbol{\eta}_{j.} = (\eta_{jt_1}, \eta_{jt_2}, \eta_{jt_3}, \eta_{jt_4})',$$

is the noise. Two sets of parameters are used with a specific role. The first term, ω_{ij}, captures the relative influence of one gene on another as compared to other genes in the putative network. The second term is the 4×4 matrix $F_{m(i)m(j)}$ which quantifies the mode of interaction and is indexed by the categorical label (1,...,4) $m(i)$, $m(j)$ of genes i and j, inferred during the previous step. Notice that matrix $F_{m(i)m(j)}$ permits to the link between genes i and j to evolves across time. This results in a global optimization criterion over the sets ω_{ij} and $F_{m(i)m(j)}$, minimizing the L_2-norm of the residuals. We then set two constraints : a) $\forall (i,j) [\![1, N_{clust}]\!]^2$, $\omega_{ij} \geqslant 0$ and b) $\forall j \in [\![1, N_{clust}]\!]$, $\sum_{i=1}^{N} \omega_{ij} \leqslant d$ where d is a non-negative parameter estimated by cross-validation. The constraints on ω_{ij} ensure that only a small number of genes will have a significant influence on any one gene leading to sparse interaction models. The second constraint is a Lasso penalty. However, no constraint is placed on the number of genes that any single gene may influence. While the full optimization is nonconvex, given the set ω_{ij} there is an analytic solution for the set $F_{m(i)m(j)}$. Similarly, given the set $F_{m(i)m(j)}$ one can solve for the set ω_{ij} via a quadratic program (QP). This leads naturally to a coordinate ascent approach. The result of the optimization is a connectivity network described by the nonzero elements of ω_{ij} combined with a set of cluster-dependent interaction models described by the set $F_{m(i)m(j)}$. Each matrix $F_{m(i)m(j)}$ is further constrained to have the following form :

$$F_{m(i)m(j)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ a_{m(i)m(j)} & 0 & 0 & 0 \\ b_{m(i)m(j)} & a_{m(i)m(j)} & 0 & 0 \\ c_{m(i)m(j)} & b_{m(i)m(j)} & a_{m(i)m(j)} & 0 \end{bmatrix} \tag{5.3}$$

where $a_{m(i)m(j)}$, $b_{m(i)m(j)}$, $c_{m(i)m(j)}$ are reals. This structure has two consequences. From a practical standpoint it reduces the complexity of the optimization from a search over 16 parameters for each $F_{m(i)m(j)}$ to one over 3 parameters. Consequently, interactions depend only on time index differences rather than absolute time index. Matrices are lower triangular with a null diagonal ; these conditions ban the possibility of feedback loop. Furthermore, as the categorical label indexes the peak in differential expression within the temporal profile we only consider causal predictor models ; that's why we impose : $m(i) \geqslant m(j) \Rightarrow F_{m(i)m(j)} = 0$. To summarize, results of the clustering are both used to select genes that are the most affected by the stimulation and to impose some constraints on the linear model. The resulting gene regulatory network is then represented by link strength ω_{ij}.

5.5.3 Simulations

In order to evaluate our inference methodology, a simulation step in which the initial gene regulatory network is perfectly known is essential for

comparison purposes. To simulate in silico data, we need to choose both a network topology and a dynamic model that spreads the signal from genes to genes. We choose two reliable network topologies : a scale free topology generated with RANGE (Long et Roth 2008) and a temporal cascade topology that represents the topology of the network when the cell is stimulated by an environmental stimulus (Yosef et Regev 2011, Alon 2007). To simulate gene expression, we assume that expression of gene A at time t depends on expression of its regulators at time $(t-1)$. To make the simulations more realistic, we used a non linear function to modelize interactions, $f(x) = \frac{C \times \exp(ax)}{b+exp(ax)}$, where a has been set to 1/3.5, b has been set to 30 and C has been set 40. This is a logistic function with a sigmoid form, classically used in modeling gene network dynamic Weaver *et al.* (1999). Futhermore, we compared our reverse-engineering method with four other algorithms : GeneNet Schafer et Strimmer (2005) based on Graphical Gaussian Models, GeneReg Huang *et al.* (2010) a regression based method that extrapolates the number of time points by B-spline regression, TD-ARACNE Zoppoli *et al.* (2010) the time course data equivalent of the information theory method ARACNE Margolin *et al.* (2006a) , and a dynamic Bayesian network method Morrissey *et al.* (2011) . We then compare the inferred matrix with the real matrix. We calculate the predictive positive value (PPV) defined as TP/(TP+FP), the sensibility defined as TP/(TP+FN), and the F-score defined as 2*sensitivity*PPV/(sensitivity+PPV) where TP represents the True Positives, FP the False Positives, and FN the False Negatives. The F-score combines both sensitivity and PPV and is known to decrease when the number of genes included in the model increases Zoppoli *et al.* (2010). We finally compute a conditional permutation test for all of these indicators of performance.

5.5.4 Microarrays, RNA interference and validation experiments

Primary microarray data were extracted from the following source Vallat *et al.* (2007). This comprises 136 samples (four time points for both unstimulated (US) and stimulated (S) cells from 6 healthy donors, 6 patients with indolent CLL and 5 patients with aggressive CLL). Patients with indolent CLL (with IGVH gene mutated and ZAP70 negative expression) had stable disease over time, while patients with aggressive CLL (4/5 with IGVH gene unmutated and 6/6 with ZAP70 positive expression) had a rapid clinical course Vallat *et al.* (2007). For the intervention experiment purpose, performed here, peripheral blood was obtained from one patient with aggressive CLL included in our previous study Vallat *et al.* (2007). B-cells were negatively selected (Rosettesep B-cell enrichment cocktail, Stemcell Biotechnologies, Vancouver, Canada) and isolated by density gradient centrifugation over Ficoll-Paque plus (Pharmacia, Upsala, Sweden). Quality of the selection was assessed by flow cytometry on a Cytomics FC500 system (Beckman-Coulter, Fullerton, CA) after CD5-PE / CD19-FITC staining (BD Biosciences, Palo Alto, CA) and was $> 98\%$. Cells were cultured at 37°C in 5% CO2 for 6 hr in RPMI 1640 medium supplemented with 10% heat inactivated FCS, 2 mM L-glutamine and 24 μg/mL gentamicin. Cells were transfected with a pool of four designed DUSP1 siRNA (siGenome SMARTpool reagent, Dharmacon Inc., USA) or with a non-sequence speci-

fic siRNA (siCONTROL non targeting siRNA#1, Dharmacon Inc., USA) at a final siRNA concentration of 100 nM using the Nucleofactor apparatus and cell line Nucleofactor kit according to the manufacturer (Amaxa Biosystem, Germany). Cells were then cultured at 37° C in 5% CO2 in supplemented RPMI-1640 culture medium. After 12 hr, the cells were recovered by density gradient centrifugation over Ficoll-Paque plus (Pharmacia, Upsala, Sweden), washed and starved for 4 hr at 37° C / 5% CO2 in supplemented RPMI 1640 medium. Starved transfected and mock-transfected B-cells at a density of 10^7cells/mL were divided in two. Half of the cells were BCR stimulated by goat F(ab')2 anti-human IgM-BIOT (Southern Biotechnology, Birmingham, AL) at 20 μg/mL and cross-linked by 20 μg/mL avidin (Sigma-Aldrich, St Louis, MO), washed and resuspended in supplemented RPMI-1640 culture medium Vallat *et al.* (2007). At four time points (60-90-210-390 min) after BCR stimulation, total mRNA was collected over four experimental conditions (DUSP1 silenced (US/S) and mock-transfected (US/S)). cRNA was prepared in accordance with the Affymetrix protocol and hybridized to the HG-HU133 plus 2.0 microarray which contains 54675 probe sets. We further normalized these 16 microarrays with the previous 136 samples with the invariant set method and the model based expression index (MBEI) obtained by the pm-mm model using dChip software Li et Wong (2001) (all data are accessible on GEO database on access number GSE39411).

5.6 Acknoledgments

We thank Dr EA. Fox (DFCI, Boston, USA), Dr E. Zorn (MGH, Boston, USA), Dr I. Delic and Dr X. Gidrol (CEA, Fontenay-aux-roses, France), Dr A. Lesne (Institut des Hautes Etudes Scientifiques, Bures-sur-yvette, France), Dr P. Georgel (Strasbourg, France), Dr P. Huneman (IHPST, Sorbonne, Paris I, France) for helpful discussions and advice, and Dr A. Rau for her critical reading of the manuscript. LV was supported in part by Fondation de France (2003004300 and 2005006549) and the Complex system Institute of Paris. This work was supported by the CLL Research Consortium PO1 CA 81538 from the National Cancer Institute (to JGG), the Eskandarian Family CLL Research Fund (to JGG), the Ligue régionale Alsace contre le Cancer, the Association pour la Recherche contre le Cancer (ARC), Fondation pour la Recherche Médicale (FRM), the Agence Nationale pour la Recherche (ANR), and LabEx TRANSPLANTEX.

6

Cascade : a R-package to study, predict and simulate the diffusion of a signal through a temporal gene network

Cette article a été publié dans la revue Bioinformatics (Jung *et al.* 2014). Dans cette article la modélisation statistique utilisée dans le Chapitre 4 est formalisée sous la forme d'une librairie additionnelle pour le logiciel R. De plus, des améliorations méthodologiques, au rang desquels le choix d'un seuillage pour les arrêtes du réseau, ainsi que des possibilités avancées de visualisations sont proposées. Des détails supplémentaires pourront être trouvés dans les Annexes B et C, qui correspondent à l'analyse de deux jeux de données à l'aide de ce package.

Les deux jeux de données analysés correspondent au jeu de données décrit dans notre Chapitre 2 et un jeu de données publié dans la littérature que nous nous sommes attachés à faire une nouvelle analyse ("E-MTAB-1475" analysé pour la première fois dans den Ham *et al.* (2013)). Notre package est capable de retrouver les résultats essentiels donnés par les auteurs, auxquels nous ajoutons des résultats supplémentaires, démontrant ainsi la valeur ajoutée de notre travail.

Les nouveautés y sont détaillées précisément. L'annexe E présente un poster détaillant les principales fonctions de cette librairie.

6.1 Abstract

Temporal gene interactions, in response to environmental stress, form a complex system that can be efficiently described using gene regulatory networks (GRN). They allow highlighting the more influential genes and spotting some targets for biological intervention experiments. Despite that many reverse-engineering tools have been designed, the Cascade package is an integrated solution adding several new and original key features such as the ability to predict changes in gene expressions after a biological perturbation in the network and graphical outputs that allow monitoring the spread of a signal through the network.

The R-package `Cascade` is available online `http:// www-math.u-strasbg.fr/genpred/spip.php?rubrique4`.

6.2 Introduction

Since the emergence of high-throughput technologies that allow measuring simultaneously expression of thousands of genes, many tools have been developed to learn gene expression profiles and reverse-engineer their underlying gene regulatory network (GRN) (Hecker *et al.* 2009, Bar-Joseph *et al.* 2012).

These tools are either based on static co-expression methods or, if the biological phenomenon shows any temporality, time dependent methods.

While the former relies on the assumption that co-expressed genes share some biological characteristics, the latter infers a directed network with temporal dependencies. In this last case, another important distinction should be made between exogenous stress (e.g., growth response) and endogenous phenomenon (e.g., cell cycle) (Zhu *et al.* 2007, Yosef et Regev 2011). This leads to different network topologies : in exogenous stress, networks' topologies seem to have larger hubs and shorter paths through temporal dependent transcriptional waves (Luscombe *et al.* 2004). This results in a quick response to environmental modifications (Luscombe *et al.* 2004). The Cascade package is designed to model such "cascade networks" taking advantage of the assignment of genes to temporal clusters, which adds temporal causality in the network.

6.3 Details on the package features

This package has been designed to analyze temporal microarray datasets, allowing gene selection, temporal cluster assignment, reverse-engineering the GRN using a penalized regression model and predicting the effect of biological intervention experiments. It also features a temporal synthetic cascade simulation tool. The biological interpretations are facilitated thanks to several graphical outputs. More insight about the statistical tools as well as benchmarks are provided in Vallat *et al.* (2013).

6.3.1 Gene selection and cluster assignment

Selecting the genes for reverse-engineering is a crucial step. Besides selecting genes with high differential expressions, the Cascade package allows enriching the selection with genes featuring specific temporal patterns. As pointed out by Hao et Baltimore (2009), several temporal gene expression waves, corresponding to specific cellular functions, can be individualized after stimulation of the cellular environment. In this pulsed biological response, some relevant genes may have low but systematic differential expressions. This selection step mostly relies on the Bioconductor R package Limma (Smyth 2005).

Each gene must be then assigned to one of the time clusters. This can be automatically performed (according to the first time when the gene is differentially expressed). Alternatively, the time clusters can be user-provided.

6.3.2 Reverse-engineering of the network

The reverse-engineering algorithm is the Lasso penalized estimation of a linear regression model described in Vallat *et al.* (2013). The Lasso penalty ensures sparsity, which is a well known feature of most biological networks (Barabási 2002). Furthermore, the temporal gene clusters are taken into account using a set of matrices $\boldsymbol{F}$ to describe how genes interact :

$$\boldsymbol{Y} = \sum_{i=1}^{N} \boldsymbol{F}_{m(X_i)m(Y)} \omega_i \boldsymbol{X}_i + \boldsymbol{\eta}, \tag{6.1}$$

where $\boldsymbol{Y}$ is the regulated gene and the $\boldsymbol{X}_i$ are potential regulator genes, the ω_i determine the strength of the link between X_i and $\boldsymbol{Y}$, $m(\cdot)$ is the function which maps a gene to its temporal cluster and $\boldsymbol{\eta}$ is a noise. Some further constraints are set to ensure a temporal causality and we use the Lasso estimator to achieve some sparsity.

It is common knowledge that biological networks are scale-free (Barabási 2002) : the distribution of the outgoing edges in the networks follows a power law distribution. As a consequence, using a statistical test (Clauset *et al.* 2009), we derived a cutoff value for the coefficients $\boldsymbol{\omega}$. It was established, by a simulation study, that such a procedure greatly improves F-scores (Van Rijsbergen 1979).

A graphical output, SI1, shows the modification of the network topology when this cutoff varies. For a given cutoff, a graphical output, SI2, shows how the stimulated transcriptional response spreads through the network. If time clusters are heterogeneous, matrices $\boldsymbol{F}$ and ω values are iteratively estimated in a coordinate ascendant approach. On the contrary, if all the time clusters are homogeneous enough, the estimation of the matrices $\boldsymbol{F}$ may be achieved using all the genes in each of the clusters, instead of using only those pointed out by their $\boldsymbol{\omega}$ values. This results in a non-iterative algorithm : matrices $\boldsymbol{F}$ and $\boldsymbol{\omega}$ values are only estimated once.

6.3.3 Prediction

We can predict changes in gene expressions, using equation (8.1), after a gene intervention experiment at the first time point, as validated, in silico and biologically in Vallat *et al.* (2013).

6.3.4 Simulation

The Cascade package provides two simulation tools. On the first hand, a random network can be simulated following the preferential attachment theory (Barabási 2002) with some constraints to ensure that the result is a temporal cascade network. On the other hand, the model, equation (8.1), can be used to simulate gene expressions from any given network.

6.4 Examples

Two package's vignettes detail the comprehensive analysis of two example datasets. A first dataset, extracted from GSE39411, is based on the transcriptional response of healthy lymphocytes B-cells after antigenic stimulation (Vallat *et al.* 2007). The second dataset (E-MTAB-1475) has a different experimental design and is based on the transcriptional response of murine lymphocytes T-cells after an in vitro stimulation that sustains cellular differentiation (den Ham *et al.* 2013). In both cases, gene expressions measured at different time points after cell stimulation are used to select genes with specific temporal patterns or high differential expressions which are then assigned to time clusters (Fig. 1 for GSE39411 and SI3-4 for E-MTAB-1475). The reverse-engineering of the GRN highlights the most influential genes in the temporal cascade (Fig. 2 and SI3-5). The impact in the GRN of a knock-down experiment of one influential gene is predicted (Fig. 3 and SI3-6).

6.5 Figures

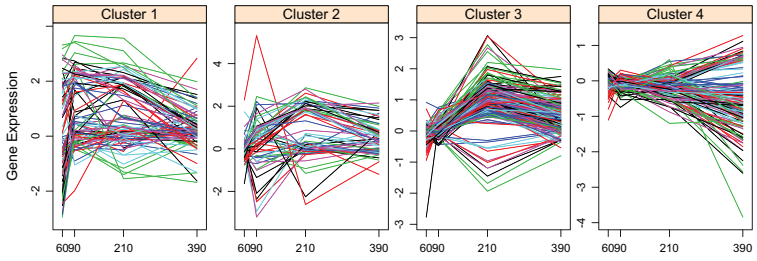


FIGURE 6.1 – *Step 1 : gene selection in GSE39411 and assignment to a time cluster.*

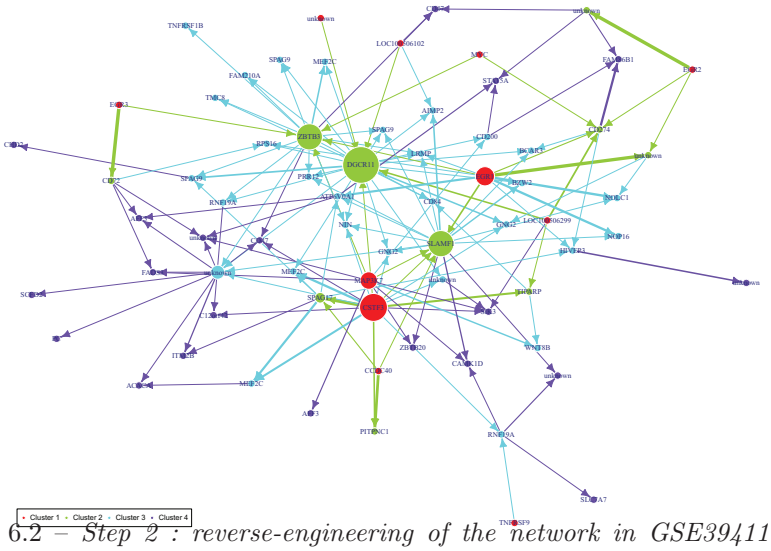


FIGURE 6.2 – *Step 2 : reverse-engineering of the network in GSE39411. Nodes represent genes and the arrows statistical links between the genes. Arrows' thickness depicts the intensity of the link.*

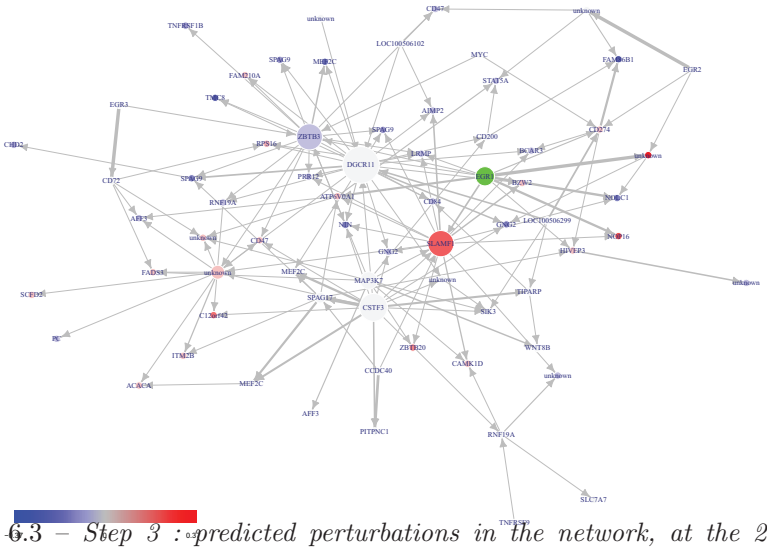


FIGURE 6.3 – *Step 3 : predicted perturbations in the network, at the 2nd time point, after gene expression modulation at an early time in the temporal GRN of GSE39411. The green influential gene is supposed to be knocked-down. Color scale legend from downregulated (blue) to upregulated (red) genes.*

7 Compléments sur le package Cascade

Nous profitons de chapitre pour résumer les principales fonctionnalités du package Cascade ainsi que les développements méthodologiques dont il est le résultat. Tout ceci est expliqué en détail dans les vignettes du package, mis à disposition dans ce manuscrit dans les annexes B et C. D'autres part, ces annexes font l'analyse de deux jeux de données différents : celui présenté dans le Chapitre 3 et traité dans les deux chapitres précédents, et le jeu de données "E-MTAB-1475" analysé pour la première fois dans den Ham *et al.* (2013).

7.1 Sélection des gènes

Dans le développement de ce package, nous avons préféré utiliser le package Limma (Smyth 2005) pour sélectionner les gènes différentiellement exprimés au détriment du modèle de mélange. La raison de ce choix est celle de l'adaptabilité. En effet, Limma est basé sur des modèles linéaires. La sélection des gènes différentiellement exprimés équivaut alors simplement au choix judicieux des bons contrastes. Notre fonction `geneSelection` fonctionne, entre autre, à l'aide d'une liste d'éléments permettant d'indiquer intuitivement ces contrastes :

- le premier élément indique soit "condition", soit "condition& time" ou "pattern". Dans le premier cas, le but de l'utilisateur sera de sélectionner les gènes différentiellement exprimés dans une condition par rapport à une autre (par exemple, stimulé versus non stimulé). L'option "condition& time" permet de sélectionner les gènes différentiellement exprimés entre deux conditions différentes à deux temps donnés. Enfin l'option "pattern" permet de sélectionner les gènes différentiellement exprimés entre deux conditions, en précisant à quels temps lesdits gènes doivent être différentiellement exprimés,
- le deuxième élément indique les deux conditions auxquels il est fait référence dans le premier élément,
- le troisième élément permet de préciser les temps de comparaison ou les patterns souhaités.

Pour de plus amples précisions, nous proposons au lecteur de consulter les vignettes des Annexes B et C ainsi que le manuel d'utilisateur du pa-

ckage.

Dans le cadre de réseau en cascade, nous avons proposé dans le Chapitre 5 une façon particulière de choisir les gènes différentiellement exprimés. En effet, l'idée naïve serait évidemment de prendre tous les gènes différentiellement exprimés entre deux conditions, mais ce procédé conduit, dans la plupart des cas, à la sélection de plusieurs milliers de gènes. Une sélection d'une telle ampleur poserait évidemment des problèmes d'ordre algorithmique. De cette liste de gènes différentiellement exprimés, nous avons voulus sélectionner ceux dont le pattern correspondait au mieux à nos attentes et à nos *a priori* biologiques. C'est pourquoi, nous proposons une sélection comme suit :

— sélectionner les gènes les plus différentiellement exprimés entre les deux conditions considérées,
— enrichir cette sélection par des gènes ayant un pic d'expression différentielle à un moment donné.

Nous avons démontré, par l'étude des fonctions de ces gènes, qu'une telle manière de les sélectionner était pertinente (voir Chapitre 5 pour le premier jeu de données et Annexe C pour le second).

Cependant, en fonction du problème considéré, toute liberté est laissée à l'utilisateur dans le choix des gènes différentiellement exprimés. Il importe cependant qu'une fois la sélection faite, l'utilisateur assigne à chaque gène un temps d'action. Dans nos travaux, nous avons toujours considéré ce temps d'action comme le premier temps où le gène est différentiellement exprimé.

7.2 Inférence du réseau

Nous ne reviendrons pas ici sur la méthodologies de l'inférence de réseau, détaillée dans les Chapitres 4 et 5. Nous traitons ici en détail un aspect de l'inférence de réseaux que nous n'avons pas encore abordé : le choix d'un seuil optimal de sélection pour les liens du réseaux.

7.2.1 Choix du seuil

Notre méthode d'inférence, est, comme nous l'avons décrit, basée sur l'utilisation de régressions pénalisée de type Lasso. Bien que les régressions Lasso permettent d'obtenir une estimation parcimonieuse du support actif de la régression (ou en d'autres termes, des régulateurs du gène considéré), il est à noter que certains des cœfficients estimés par le Lasso peuvent être très faible en valeur absolue. Cela conduit à l'estimation d'un réseau certes parcimonieux, mais avec quantités de liens si faibles que dénués d'intérêt (ne serait-ce qu'en terme de prédiction). Pour fixer les choses, nous tirons de la vignette présentée dans l'Annexe C un réseau pour lequel aucun seuillage n'est effectué (Figure 7.1). Nous présentons ensuite ce même réseau avec un choix optimal de seuillage (Figure 7.2).

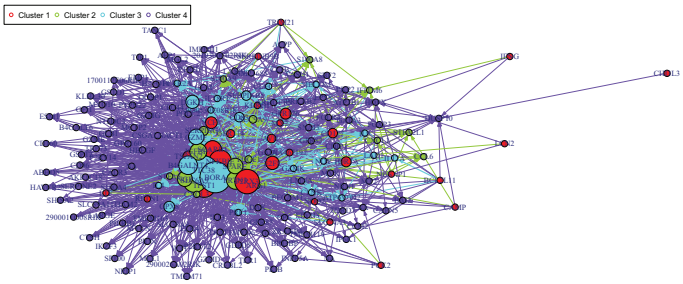


FIGURE 7.1 – *Réseau sans seuillage : il y a une forte densité de liens, dont certains sont très proches de la nullité.*

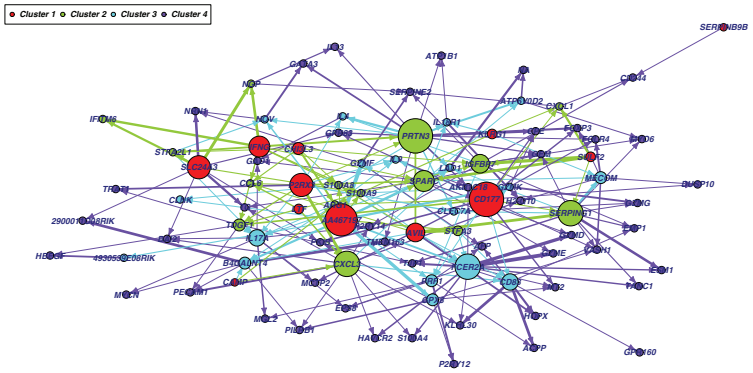


FIGURE 7.2 – *Réseau avec seuillage : la structure principale du réseau est apparente.*

La question qui suit est : comment choisir ce seuil de façon optimale ? Dans le package Cascade nous proposons une approche en deux temps. Dans un premier temps, il s'agit de regarder l'évolution du réseau en fonction de l'augmentation du seuil. Quel intérêt ? Il s'agit de vérifier que la structure du réseau reste semblable et robuste au choix du seuil. Autrement dit, nous souhaiterions que les hubs, ces gènes fortement régulateurs, gardent une position centrale dans le réseau quelque soit le seuil choisi. Dans tous les cas que nous avons eu à traiter, cette propriété était effectivement vérifiée. Si tel n'est pas le cas, un examen approfondi du réseau est nécessaire.

Une fois cette vérification faite, il est possible de choisir le seuil optimal. Nous avons choisi de déterminer cette optimalité dans le sens où nous souhaitions choisir le seuil permettant d'avoir le réseau le plus proche possible d'un réseau invariant d'échelle. Pour ce faire, nous avons utilisé un test d'adéquation aux lois de type puissance (celles-la même qui sont les distributions théoriques du nombre de liens régulés) (Clauset *et al.* 2009).

Les auteurs indiquent qu'une valeur p pour leur test au-dessus de 0,10 est un bon indicateur d'une loi suivant une distribution de type puissance. Cela laisse souvent le choix entre plusieurs seuils possibles. Une étude par simulation sur 500 réseaux à permis de déterminer des zones à choisir préférentiellement (voir Figure 7.3).

Nous avons ensuite testé cette procédure de choix de seuil par des simu-

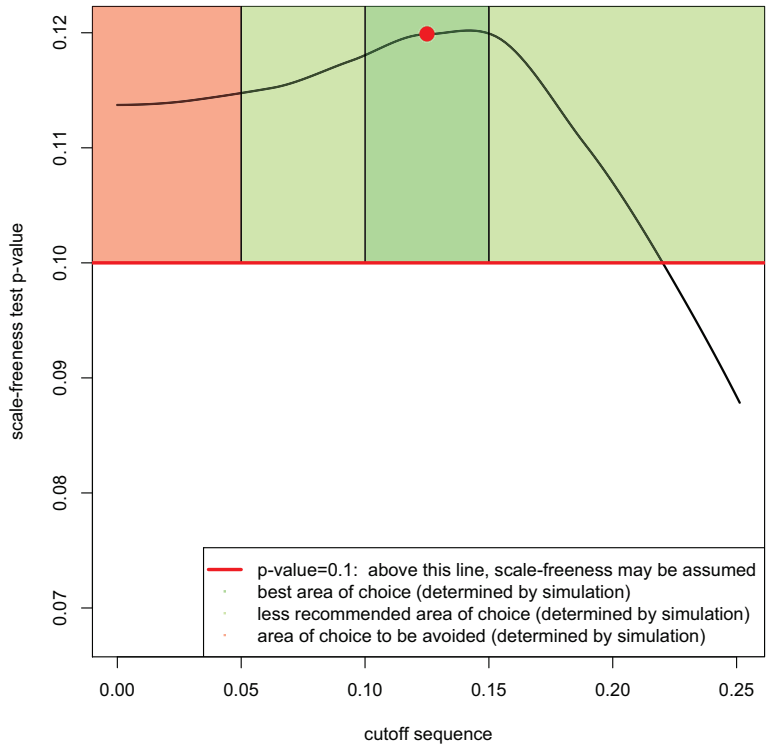


FIGURE 7.3 – *Choix du seuil en fonction de la valeur p du test d'adéquation à une distribution de type puissance. Les différentes zones de choix préférentiel permettent de choisir le seuil optimal parmi tous les seuils résultants en une valeur p supérieure à 0,1.*

lations. Nous avons choisi une grille de seuil allant de 0 à 0,50 avec un pas de 0,01. Pour chacun de ces seuils nous avons calculé le Fscore correspondant. Nous avons par ailleurs calculé le Fscore obtenu par le choix du seuil déterminé par notre procédure. Nous avons finalement calculé le ratio du Fscore optimal sur le Fscore obtenu par notre procédure. Nous avons obtenu un ratio non significativement différent de 1. La variance de ces ratios étaient de 0,1. Notre procédure ne choisit donc pas systématiquement un seuil trop petit ou trop grand et apparaît comme étant un choix judicieux de seuillage.

Le package permet en outre d'analyser le réseau obtenu, en calculant différent indicateur sur les gènes. Ces indicateurs ont été défini dans le Chapitre 1. Ils permettent de déterminer les gènes importants, en donnant à cette notion d'importance plusieurs nuances. Un point important est que les réseaux inférés sont pondérés. Il faut donc utiliser une méthodologie adaptée pour calculer les différents indicateurs (Opsahl *et al.* 2010) (voir Annexe B et C). En outre le package permet de réaliser des prédictions des expériences d'intervention sur les gènes (en particulier, il est possible de reproduire des prédictions comme celles faites pour DUSP1 dans le Chapitre 5). Enfin, le package permet de reproduire l'ensemble des simulations de réseaux et d'expressions de gènes mentionnées dans ce chapitre.

Le chapitre suivant se consacre à améliorer méthodologiquement la régression Lasso, afin d'éliminer automatiquement tous les liens non robustes.

Grâce à cela, nous ne serons plus obligé de passer par la procédure de choix de seuil que nous venir de définir.

8 selectBoost : a general algorithm to enhance the performance of variable selection methods

Ce chapitre correspond à un article qui n'a pas encore été publié. Le but de ce dernier est de présenter un algorithme permettant d'améliorer la précision des méthodes de sélection de variable (c'est-à-dire la proportion de variables sélectionnées à raison). Derrière ce travail se cache l'idée d'une meilleure sélection des régulateurs des gènes lorsque nous inférons les réseaux de régulation de gènes. Une telle méthode permet également de ne plus avoir à seuiller les liens les plus faibles. L'annexe D présente les figures complémentaires citées.

8.1 Abstract

Variable selection has become one of the major challenge in statistics. Although lots of methods have been proposed in the literature their performance in terms of recall and precision are limited in a context where the number of variables exceed from far the number of observations or in a high correlated setting. In this article, we propose a quite general algorithm which can improve the precision of any existing variable selection method. This algorithm is based on highly intensive simulations and takes into account the correlation structure of the data. Our algorithm can either produce a confidence index or be used in a experimental design planning perspective. We demonstrate the performance of our algorithm on both simulated and real data and we show its adaptability.

8.2 Introduction

The problem of variable selection has received an increasing attention over the last years (Fan et Li 2006) and is one of the most important

challenges for the 21st century (Donoho *et al.* 2000). Indeed, technological innovations make it possible to measure large amounts of data relatively low cost. As a consequence, problems in which the number P of variables is greater than the number N of observations have become common. As reviewed by Fan and Li (Fan et Li 2006), such situations arise in many fields from sciences to humanities, and variable selection may be of great help to answer challenges that are specific to each of them. For example, in biology, thousands of messenger RNA (mRNA) gene expressions (Lipshutz *et al.* 1999) may be potential predictors of some illness. Other examples are imagery (magnetic resonance image, nuclear magnetic resonance, satellite images...), financial engineering and risk management or health studies (Fan et Li 2006). Moreover, in such studies, the correlation between variables is often very strong (Segal *et al.* 2003) and variable selection methods often fail to choose the informative variables among those which are not. Here we propose a general algorithm that improves the performances of any existing variable selection method.

In this article, we assume that our data is generated by a multivariate linear model :

$$\boldsymbol{y} = \mu \mathbf{1}_N + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{8.1}$$

where $\boldsymbol{y} = (y_1, ..., y_N)'$ is the response variable, μ is the mean variable response, $\mathbf{1}_N$ is a vector of length N containing only ones, $\boldsymbol{X} = (\boldsymbol{x}_{1.}, ..., \boldsymbol{x}_{P.})$ is the design matrix of size $N \times P$, $N > 2$, with $\boldsymbol{x}_{p.} = (x_{p1}, ..., x_{pN})'$ which are the variables and $\boldsymbol{\varepsilon} = (\varepsilon_1, ..., \varepsilon_N)$ is a Gaussian noise vector which is the realization of some random law with a mean of 0 and an unknown variance σ^2. Furthermore, we will assume that the vector of parameters $\boldsymbol{\beta} = (\beta_1, ..., \beta_P)'$ is sparse. In other words, we will assume that $\beta_i = 0$ except for a quite small proportion of elements of the vector. We note $\mathcal{S}$ as the set of indexes for which $\beta_i \neq 0$ and $q < \infty$ is the cardinal of this set $\mathcal{S}$. Without any loss of generality, we will assume that $\beta_p \neq 0$ if and only if $p \leqslant q$. Moreover, we assume that the response and the variables are centred and that $\|\boldsymbol{x}_{p.}\|^2 = 1$ for $p = 1, ..., P$ where $\|\cdot\|$ stands for the usual euclidean norm ; in this context, we have $\mu = 0$.

When dealing with a problem of variable selection, there are three main goals. We enumerate them in increasing level of difficulty :

1. The prediction goal, in which you want $\hat{\boldsymbol{y}}$ to be as close as possible to $\boldsymbol{y}$.
2. The estimation goal, in which you want $\hat{\boldsymbol{\beta}}$ to be as close as possible to $\boldsymbol{\beta}$.
3. The estimation of the support, in which you want $\mathbb{P}(\mathcal{S} = \hat{\mathcal{S}})$ to be close to one.

Fan and Li (Fan et Li 2001) proposed another desirable property, the oracle property, which combines goals 2 and 3. Precisely, a method is said to have the oracle property if it discovers the correct support, and if the rate of convergence of $\hat{\boldsymbol{\beta}}$ toward $\boldsymbol{\beta}$ is optimal (*i.e.* the same as in the case in which the correct support is known). Here, our interest is mainly in

the third goal, *i.e.* in identifying the correct support $\mathcal{S}$. This kind of issue arises in many fields, for example in biology, where it is of greatest interest to discover which specific molecules are involved in a disease (Fan et Li 2006).

There is a vast literature dealing with the problem of variable selection in both statistical and machine learning areas ((Fan et Li 2006, Fan et Lv 2010)). The main variable selection methods can be gathered in the common framework of penalized likelihood. The estimate $\hat{\boldsymbol{\beta}}$ is then given by :

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}\in\mathbb{R}^P} \left[-\ell_N(\boldsymbol{\beta}) + \sum_{p=1}^{P} \text{pen}_{\boldsymbol{\lambda}}(\beta_p) \right], \tag{8.2}$$

where $\ell_N(.)$ is the log-likelihood function, $\text{pen}_{\boldsymbol{\lambda}}(.)$ is a penalty function with k parameters and $\boldsymbol{\lambda} = (\lambda, \lambda_2, \lambda_3, ..., \lambda_k)'$. As the goal is to obtain a sparse estimation of the vector of parameters $\boldsymbol{\beta}$, a natural choice for the penalty function is to use the so-called L_0 norm ($\|.\|_0$) which corresponds to the number of non-vanishing elements of a vector :

$$\begin{array}{rccc} \text{pen}_{\boldsymbol{\lambda}} : & \mathbb{R} & \to & \{0, \lambda\} \\ & x & \mapsto & \left\{ \begin{array}{ll} \text{pen}_{\boldsymbol{\lambda}}(x) = \lambda & \text{if } x \neq 0 \\ \text{pen}_{\boldsymbol{\lambda}}(x) = 0 & \text{else} \end{array} \right. \end{array} \quad \Rightarrow \sum_{p=1}^{P} \text{pen}_{\boldsymbol{\lambda}}(\beta_p) = \lambda \|\boldsymbol{\beta}\|_0 . \tag{8.3}$$

For example, when $\lambda = 1$, we get the Akaike Information Criterion (AIC) (Akaike 1974) and when $\lambda = \frac{\log(N)}{2}$ we get the Bayesian Information Criterion (BIC) (Schwarz 1978). Another slightly different formulation leads to Mallow's C_p (Mallows 1973) or to the Risk Inflation Criterion (Foster 1994). In the context of Gaussian independent and identically distributed (i.i.d.) errors in the model described in equation (8.1), the following holds (Burnham et Anderson 2002) :

$$-\ell_N(\boldsymbol{\beta}) = \frac{N}{2} \log \left(\frac{\|\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\|^2}{N} \right) + K_1, \tag{8.4}$$

where K_1 is a constant. Up to an affine transformation of the log-likelihood (Fan et Lv 2010), we see that equation (8.2) is equivalent to :

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}\in\mathbb{R}^P} \left[\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \sum_{p=1}^{P} \text{pen}_{\boldsymbol{\lambda}}(\beta_p) \right]. \tag{8.5}$$

A lot of different penalties can be found in the literature. Solving this problem with $\|.\|_0$ as part of the penalty is an NP-hard problem (Natarajan 1995, Fan et Lv 2010). It cannot be used in practice when P becomes large, even when it is employed with some search strategy like forward regression, stepwise regression(Hocking 1976), genetic algorithms (Koza *et al.* 1999)... Donoho and Elad (Donoho et Elad 2003) show that relaxing $\|.\|_0$ to norm $\|.\|_1$ ends, under some assumptions, to the same estimation. This result encourages the use of a wide range of penalty based on different norms. For example, the case where $\text{pen}_{\boldsymbol{\lambda}}(\beta_p) = \lambda|\beta_p|$ is the Lasso estimator (Tibshirani 1996) (or equivalently Basis Pursuit Denoising (Chen *et al.* 2001)) whereas $\text{pen}_{\boldsymbol{\lambda}}(\beta_p) = \lambda\beta_p^2$ leads to the Ridge estimator (Hoerl et Kennard

1970). These two last cases can be seen as a special case of Bridge regression (Frank et Friedman 1993) in which $\text{pen}_{\lambda}(\beta_p) = \lambda|\beta_p|^b$ with $0 < b \leqslant 2$. Nevertheless, the penalty term induces variable selection only if :

$$\min_{x \geqslant 0} \left(\frac{\text{dpen}_{\lambda}(x)}{\text{d}x} + x \right) > 0. \tag{8.6}$$

This explains why the Lasso regression allows variable selection while the Ridge regression does not. As it is well known (Zou 2006), the Lasso leads to a biased estimate. The SCAD (smoothly clipped absolute deviation) (Fan 1997), MCP (minimax concave penalty) (Zhang 2010) or adaptative Lasso (Zou 2006) penalties all address this problem. The popularity of such variable selection methods is linked to fast algorithms like LARS (least-angle regression) (Efron *et al.* 2004), coordinate descent (Wu et Lange 2008) or PLUS (Zhang 2010).

Nevertheless, the goal of identifying the correct support of the regression is complicated and the reason why variable selection methods fail to select the set of non-zero variables $\mathcal{S}$ can be summed up in one word : linear correlation. Choosing the Lasso regression as a special case, Zhao and Yu (Zhao et Yu 2006) (and simultaneously Zou (Zou 2006)) found an almost necessary and sufficient condition for Lasso sign consistency (*i.e.* selecting the non-zero variables with the correct sign). This condition is known as "irrepresentable condition" :

$$\left| \boldsymbol{X}'_{\setminus \mathcal{S}} \boldsymbol{X}_{\mathcal{S}} \left(\boldsymbol{X}'_{\mathcal{S}} \boldsymbol{X}_{\mathcal{S}} \right)^{-1} \text{sgn}(\boldsymbol{\beta}_{\mathcal{S}}) \right| < \mathbf{1}, \tag{8.7}$$

where $\boldsymbol{X}_{\mathcal{S}} = (x_{ij})_{i,j \in \mathcal{S}^2}$, $\boldsymbol{\beta}_{\mathcal{S}} = (\beta_p)_{p \in \mathcal{S}}$. In other words, when $\text{sgn}(\boldsymbol{\beta}_{\mathcal{S}}) = \mathbf{1}_q$, this can be seen as the regression of each variable which is not in $\mathcal{S}$ over the variables which are in $\mathcal{S}$. As all variables in the matrix $\boldsymbol{X}$ are centered, the absolute sum of the regression parameters should be smaller than 1 to satisfy this "irrepresentable condition".

Facing this issue, existing variable selection methods can be split into two categories :

— Those which are "regularized" and try to give a similar coefficients to variables which are correlated (*e.g.* : elastic net (Zou et Hastie 2005)),

— Those which are not "regularized" and pic up one variable among a set of correlated variables (*e.g.* : the Lasso (Tibshirani 1996)).

The former group can then be split into methods in which groups of correlation are known, such as the group Lasso (Yuan et Lin 2006, Friedman *et al.* 2010) and those in which groups are not known as in the elastic net (Zou et Hastie 2005). The latter combines the $\mathcal{L}_1$ and the $\mathcal{L}_2$ norm and takes advantage of both. Broadly speaking, non-regularized methods will select some co-variables among a group of correlated variables while regularized methods will select all variables in the same group with similar coefficients (see example in Figure 2).

However, none of these selection methods distinguishes between variables that were selected for inclusion in the model with confidence and

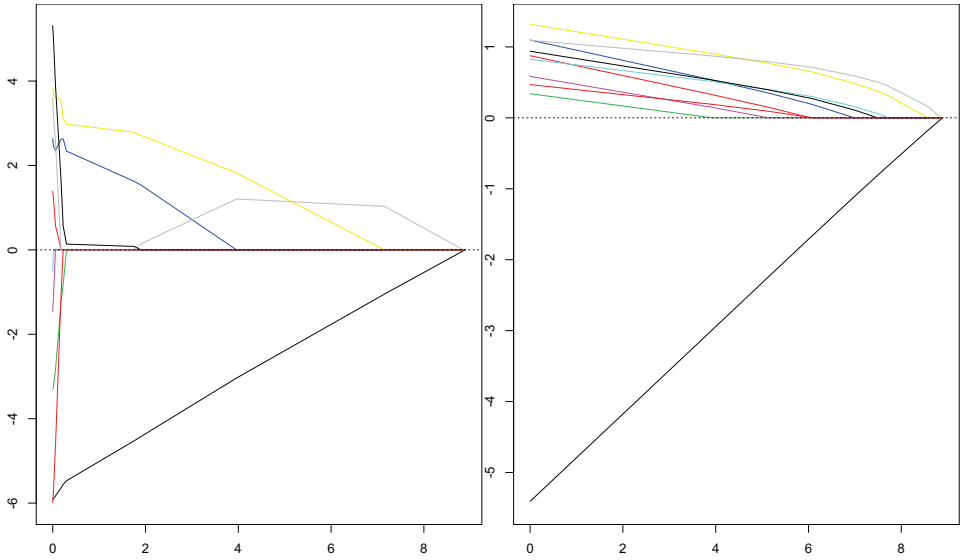


FIGURE 8.1 – *In this example* $N = 20, P = 10$, $\boldsymbol{\beta} = (1,1,0,...,0)'$. *The mean correlation between* $\boldsymbol{x}_{1.}$ *and the other variables is* 0.20 *while the mean correlation between all the other variables is* 0.95. *The x-axis corresponds to the value of the penalty parameter* λ *; the greater the parameter, the stronger the constraint. Left : with the lasso regression, no regularization is made. Right : with the elastic net regression, the coefficients of correlated variables are similar.*

those that were not. In this article, we propose the selectBoost algorithm that can provide a confidence factor for selected variables. Our new algorithm will be useful in different contexts, including biology where it will allow high precision selection of relevant therapeutic targets.

The rest of this article is organized as follows. In section 2 we present our new algorithm, in section 3 we drive some simulation studies. A real dataset will be analyzed in section 4, while section 5 will end with some remarks and conclusion notes.

8.3 METHODS

The selectBoost algorithm has been designed in a general framework whose goal is to enhance the abilities of any variable selection method, especially those which are not regularized. The main goal is to improve the precision, *i.e.* the proportion of selected variables which really are in $\mathcal{S}$.

8.3.1 Introduction

The main idea of our algorithm is to consider that groups of variables of the matrix $\boldsymbol{X}$ which are linearly correlated are independent realizations of the same random function. According to this random function, correlated variables are then perturbed. Strictly speaking, the use of noise to determine the informative variables is not a new idea. For example, it has been shown that adding random pseudo-variables decreases over-fitting (Wu et Stefanski 2007). In the case where $P > N$ the pseudo-variables are generated either with independent Gaussian laws $\mathcal{N}(0,1)$ or by using permutations on the matrix $\boldsymbol{X}$ (Wu et Stefanski 2007). Another approach

consists in adding noise to the response variable and leads to similar results (Luo *et al.* 2006). The rational of this last method is based on the work of Cook and Stefanski (Cook et Stefanski 1994) which introduces the simulation-based algorithm SIMEX (Cook et Stefanski 1994). Adding noise to the matrix $\boldsymbol{X}$ has already been used in the context of microarrays (Chen *et al.* 2007). Simsel (Eklund et Zwanzig 2012) is an algorithm that both adds noise to variables and uses random pseudo-variables. One new and interesting approach is stability selection (Meinshausen et Bühlmann 2010) in which the variable selection method is applied on sub-samples, and informative variables are defined as variables which have a high probability of being selected. Bootstraping has been applied to the Lasso on both response variable and the matrix $\boldsymbol{X}$ with better results in the former case (Bach 2008). The random Lasso, in which variables are weighted with random weights, has also been introduced (Wang *et al.* 2011).

In this article, following the idea of using simulation to enhance the variable selection methods, we propose the selectBoost algorithm. Unlike other algorithms reviewed above, our algorithm takes care of the correlation structure of the data. Furthermore, our algorithm is motivated by the fact that in the case of non-regularized variable selection methods, if a group contains variables that are highly correlated together, one of them will be chosen "at random" (Zou et Hastie 2005).

As we assume that the variables are centred and that $\|\boldsymbol{x}_{p.}\|^2 = 1$ for $p = 1, ..., P$, we know that $\boldsymbol{x}_{p.} \in \mathcal{S}^{N-2}$. Indeed, the normalization puts the variables on the unit sphere $\mathcal{S}^{N-1}$. The process of centring can be seen as a projection on the hyperplane $\mathcal{H}^{N-1}$ with the unit vector as normal vector. Moreover, the intersection between $\mathcal{H}^{N-1}$ and $\mathcal{S}^{N-1}$ is $\mathcal{S}^{N-2}$. We further define the following isomorphism :

$$\begin{array}{rccl} \phi & : & \mathcal{H}^{N-1} & \rightarrow & \mathbb{R}^{N-1} \\ & & \boldsymbol{h_n} & \mapsto & \phi(\boldsymbol{h_n}) = \boldsymbol{f_n} \quad n = 1, ..., N-1, \end{array} \tag{8.8}$$

where $\{\boldsymbol{h_n}\}_{n=1,...,N-1}$ is an orthogonal base of $\mathcal{H}^{N-1}$ and $\{\boldsymbol{f_n}\}_{n=1,...,N-1}$ is the canonical base of $\mathbb{R}^{N-1}$. We can define the following orthogonal base of $\mathcal{H}^{N-1}$:

$$\boldsymbol{h_n} = \frac{\sum_{i=1}^{n} \boldsymbol{e_i} - (n-1)\boldsymbol{e_{n+1}}}{\| \sum_{i=1}^{n} \boldsymbol{e_i} - (n-1)\boldsymbol{e_{n+1}} \|},$$

with $\{\boldsymbol{e_n}\}_{n=1,...,N}$ the canonical base of $\mathbb{R}^N$. Note that $\phi(\mathcal{S}^{N-2}) = \mathcal{S}^{N-2}$, and that is why we can work in $\mathbb{R}^{N-1}$ and then return in $\mathbb{R}^N$.

8.3.2 The selectBoost algorithm

To use the selection-boost algorithm, we need a grouping method gr_{c_0} depending on an user-provided constant $0 \leqslant c_0 \leqslant 1$. This constant determines the strength of the grouping effect. The grouping method maps each variable index $1, ..., P$ to an element of $\mathcal{P}(\{1, ..., P\})$ (with $\mathcal{P}(S)$ is the power-set of the set S, *i.e.* the set which contains all the subsets of S.) . Concretely, $gr_{c_0}(p)$ is the ensemble of all variables which are considered to be linked to

the variable $\boldsymbol{x}_p$ and $\boldsymbol{X}_{gr_{c_0}(p)}$ is the submatrix of $\boldsymbol{X}$ containing the columns which indices are in $gr_{c_0}(p)$. We impose the following constraints to the grouping function :

$$\forall p \in \{1, ..., P\} : \quad gr_1(p) = \{p\} \quad \text{and} \quad gr_0(p) = \{1, ..., P\}. \tag{8.9}$$

Furthermore, we need to have a selection method $select : \mathbb{R}^{N \times P} \times \mathbb{R}^N \rightarrow \{0,1\}^P$ which maps the design matrix $\boldsymbol{X}$ and the response variable $\boldsymbol{y}$ to a 0-1 vector of length P with 1 at position p if the method selects the variable p and 0 otherwise.

Here, we make the assumption that a group of correlated variables are independent realizations of the same multivariate Gaussian law. As the variables are normalized with respect to the $\mathcal{L}_2$ norm, we will use the von-Mises Fisher law (Sra 2012) in $\mathbb{R}^{N-1}$ thanks to the isomorphism ϕ. The probability density function of the von Mises-Fisher distribution for the random P-dimensional unit vector $\boldsymbol{x}$ is given by :

$$f_P(\boldsymbol{x}; \boldsymbol{\mu}, \kappa) = \tilde{K}_P(\kappa) \exp\left(\kappa \boldsymbol{\mu}' \boldsymbol{x}\right)$$

where $\kappa \geqslant 0$, $\boldsymbol{\mu} = (\mu_1, ..., \mu_P)'$, $\|\boldsymbol{\mu}\|_2 = 1$ and the normalization constant $C_P(\kappa)$ is equal to

$$\tilde{K}_P(\kappa) = \frac{\kappa^{P/2-1}}{(2\pi)^{P/2} I_{P/2-1}(\kappa)},$$

where I_v denotes the modified Bessel function of the first kind and order v (Abramowitz et Stegun 1972).

We then use the von-Mises Fisher law to create replacement of the original variables by some simulations (see Algorithm 1) to create B new design matrices $\boldsymbol{X}^{(1)}, ..., \boldsymbol{X}^{(B)}$. The selectBoost algorithm then applies the variable selection method *select* to each of these matrices and returns a vector of length P with the frequency of apparition of each variable. The frequency of apparition of variable $\boldsymbol{x}_{p.}$, noted ζ_p is assumed to be an estimator of the probability $\mathbb{P}(\boldsymbol{x}_{p.} \in \mathcal{S})$ for this variable to be in $\mathcal{S}$. Nevertheless, both the grouping method and the choice of c_0 are crucial. When this constant is too small, the model is not perturbed enough. On the other hand, when this constant is too large, variables are chosen at random.

The selectBoost algorithm returns the vector $\boldsymbol{\zeta} = (\zeta_1, ..., \zeta_P)'$. One has now to choose a threshold to determine which variables are selected. In this article, we choose to select a variable p if $\zeta_p = 1$. In some applications, lower choices of threshold may be chosen.

8.3.3 Choosing the parameters of the algorithm

We first have to choose the grouping function. One of the simplest way to define a grouping function gr_{c_0} is the following :

Algorithme 1 : Pseudo-code for the selectBoost algorithm with c_0 fixed

Require: $gr_{c_0}, select, B, c_0, P$
  $\zeta \leftarrow \mathbf{0}_P$
  for $b = 1, ..., B$ **do**
    $\boldsymbol{X}^{(b)} \leftarrow \boldsymbol{X}$
    for $p = 1, ..., P$ **do**
      $\boldsymbol{x}_{p.}^{(b)} \leftarrow \phi^{-1}\left(\text{random-vMF}\left(\hat{\boldsymbol{\mu}}(\phi(\boldsymbol{X}_{gr_{c_0}(p)})), \hat{\kappa}(\phi(\boldsymbol{X}_{gr_{c_0}(p)})\right)\right)$
    end for
    $\zeta \leftarrow \zeta + select(\boldsymbol{X}^{(b)}, \boldsymbol{y})$
  end for
  $\zeta \leftarrow \zeta / B$

$$gr_{c_0}(p) = \left\{ p' \in \{1, ..., P\} \middle| \; | < \boldsymbol{x}_{p.}, \boldsymbol{x}_{p'.} > | \geqslant c_0 \right\}. \tag{8.10}$$

In other words, the correlation group of the variable p is determined by variables whose correlation with $\boldsymbol{x}_{p.}$ is at least c_0. In the following this method will be refereed as the "naive" grouping method. Nevertheless, the structure of correlation may further be taken into account using graph community clustering. Let $\boldsymbol{C}$ be the correlation matrix of matrix $\boldsymbol{X}$. Let define $\check{\boldsymbol{C}}$ as follows :

$$\check{c}_{ij} = \begin{cases} |\check{c}_{ij}| & \text{if} \quad |\check{c}_{ij}| > c_0 \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases}.$$

Then, we apply a community clustering algorithm on the undirected network with weighted adjacency matrix defined by $\check{\boldsymbol{C}}$.

One the grouping function chosen we have to choose parameter c_0. Due to the constraints in equation (8.9) the selectBoost algorithm results in the initial variable selection method when $c_0 = 1$. As we will show in the next session, the smaller the c_0 parameter, the higher the precision of the resulting selected variables. On the other hand, it is obvious that the probability of choosing none of the variables (*i.e.* resulting in the choice of the empty set) increases as the parameter c_0 decreases. In the perspective of experimental planning, the choice of c_0 should result of a compromise between precision and proportion of empty models. Nevertheless, the c_0 parameter can be used to introduce a confidence index γ_p related to the variable $\boldsymbol{x}_{p.}$:

$$\gamma_p = 1 - \min_{\boldsymbol{x}_{p.} \in \hat{\mathcal{S}}_{c_0}} c_0. \tag{8.11}$$

It should be noted that the confidence index values 0 if the variable is not chosen initially by the variable selection method *select* and $0 \leqslant \gamma_p \leqslant 1$.

8.4 Numerical studies

8.4.1 Introduction

To access the performances of the selectBoost algorithm, we performed numerical studies. As stated before, the selectBoost algorithm can be applied to any existing variable selection method. Here, we decided to use the Lasso and forward stepwise selection. The performance of the Lasso is known to be strongly dependant on the choice of the penalty parameter λ. In our simulations, we used four criteria to choose this penalty parameter : BIC, modified BIC (BIC2) in which the estimation of the residual variance is calculated with the model including two variables, AICc which is known to be asymptotically equivalent to cross-validation and generalized cross-validation (GCV).

To demonstrate the performance of the selectBoost method, we compared our method with stability selection and with a naive version of our algorithm, naiveSelectBoost. The naiveSelectBoost algorithm works as follows : estimate $\boldsymbol{\beta}$ with any variable selection method then if $gr_{c_o}(p)$, as defined in equation (8.10) for example, is not reduced to $\{p\}$, shrink $\hat{\beta}_p$ to 0. The naiveSelectBoost algorithm is similar to the selectBoost algorithm, except that it does not take into account the error which is made choosing at random a variable among a set of correlated variables.

We explored four situations. Let P be the number of variables and N the number of observations. Data are generated from model in equation (8.1), assuming that $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. The variance σ^2 is chosen to reach a signal to noise ratio of 5. Exception made of situation 4, variables are simulated following a multivariate Gaussian law, with variance-covariance matrix Σ. The diagonal elements of Σ are always set to 1. Each situation is repeated 200 times.

Situation 1 We are in the case where $P = N = 10$ and $\boldsymbol{\beta} = (1, 1, 1, 0, 0, ..., 0)'$. We set $\Sigma_{ij} = 0$ for $1 \leqslant i \neq j \leqslant 9$ and $\Sigma_{1,10} = 0$.

Situation 2 We are in the case where $P = 50$ and $N = 20$ and $\boldsymbol{\beta} = (1, 1, 1, 1, 1, 0, 0, ..., 0)'$. We set $\Sigma_{ij} = 0.5$ for $1 \leqslant i \neq j \leqslant 50$.

Situation 3 We are in the case where $P = 500$ and $N = 25$ and $\boldsymbol{\beta} = (1, 1, 1, 1, 1, 0, 0, ..., 0)'$. We set $\Sigma_{ij} = 0.5$ for $1 \leqslant i \neq j \leqslant 500$.

Situation 4 In this situation we use gene expression from a microarray data experiment in which $N = 24$. We first select the 1300 genes that were differentially expressed (stimulated versus unstimulated). For each repetition, we randomly select 100 genes among the 1300 and use the model in equation (8.1) to generate the response variable. We set $\boldsymbol{\beta} = (1, 1, 1, 1, 1, 0, 0, ..., 0)'$.

We use 4 indicators to evaluate the abilities of our method on simulated data. We define :

— recall as the ratio of the number of correctly identified variables (*i.e.*

$\hat{\beta}_i \neq 0$ and $\beta_i \neq 0$) over the number of variables that should have been discovered (*i.e.* $\beta_i \neq 0$).

— precision as the ratio of correctly identified variables (*i.e.* $\hat{\beta}_i \neq 0$ and $\beta_i \neq 0$) over the number of identified variables (*i.e.* $\hat{\beta}_i \neq 0$).

— Fscore as the following ratio :

$$2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}.$$

— emptiness as the proportion of empty models (no variable is selected)

Recall, precision, and Fscore are calculated over all models that are not empty. Note that our interest is focused on precision, as our goal is to select reliable variables. When $c_0 = 1$ the selectBoost algorithm has no difference with the initially selected method *select*. When c_0 is decreasing toward zero we expect a profit in precision and a decrease of recall. We also calculate the Fscore which combines both recall and precision. As an improvement of precision comes with an increase of the proportion of empty models, the best method is one with the highest precision for a given level of emptiness.

8.4.2 Results of the simulation

Only an extract of the results is presented in the main part of the article ; full results are available in supporting informations. We first analyze the results for each pair of selection method and situation. We show the evolution of the four criteria (precision, recall, Fscore and emptiness) in function of the decrease of c_0. When $c_0 = 1$, the selectBoost algorithm is equivalent to the initial variable selection method. As our main focus is on precision, we add three histograms representing the evolution of the precision distribution for the highest, an intermediate and the lowest c_0. Figure 8.2 shows the result for the Lasso with the modified BIC in Situation 1. In this example, we succeed to improve precision from 0.63 to 0.93. Other variable selection methods show interesting improvement of precision : the gain in precision is the lowest for the Lasso with the BIC criterion. This is not surprising since this method reaches the highest level of precision when $c_0 = 1$. On the other hand, the Lasso with AICc or GCV present the greatest improvement in precision with the decrease of c_0 : in Situation 2, for the Lasso with GCV, precision improves from 0.25 to 0.75. However, as shown by the histograms of the precision, the proportion of models for which precision reaches one increases with the decrease of c_0. The Fscore remains either stable or shows a small decrease indicating that the loose in recall is compensated by the increase of precision. In other words, our method allows to choose the desired trade-off between recall and precision.

As our interest is focused on precision, our goal is to reach the highest precision with the lowest proportion of empty models. In this context, one interesting fact about the selectBoost method is that the method of choice of the penalty parameter in the Lasso is no more crucial. Indeed, as shown in Figure 8.3, the precision of each method is similar at a given proportion of emptiness. Nevertheless, depending of the situation, the choice of the penalty parameter by AICc (see Annex D) or GCV (see Figure 8.3) may

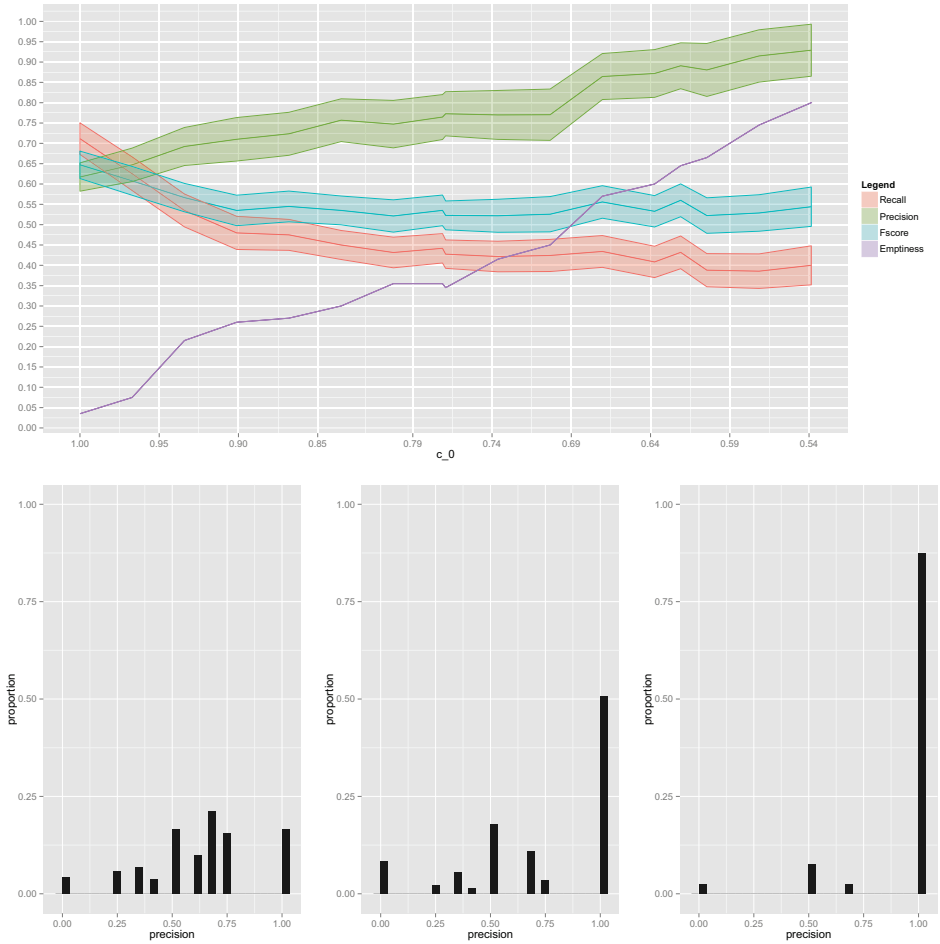


FIGURE 8.2 – *Example of result, here Situation 1 with the Lasso with the modified BIC criterion. Top figure : evolution of the four indicators (recall, precision, Fscore and emptiness) with 95% bandwidth confidence interval in function of* c_0*. Bottom : the distribution of the precision among all non-empty models for the highest, an intermediate, and the lowest* c_0*.*

lead to worse outcomes, even if there is an increase of precision with the confidence index.

Except in one case (the Lasso with choice of penalty parameter through the BIC criterion, see Figure 8.3), the selectBoost algorithm shows its superiority over the naiveSelectBoost algorithm. The error which is made when choosing randomly a variable among a set of correlated variables conduces to further wrong choice of variables. While the intensive simulation of our algorithm allows to take into account this error, the naiveSelectBoost does not. The superiority of the naive algorithm in Situation 1 for the Lasso with BIC criterion may be the consequence of the small size of the data and the low correlation setting.

Finally we compare the selectBoost algorithm with stability selection. Stability selection use a re-sampling algorithm to determine which of the variable which are included in the model are robust. In our simulation, stability selection shows performances with relative high precision but also high proportion of empty models. Moreover, in contrast to the select-

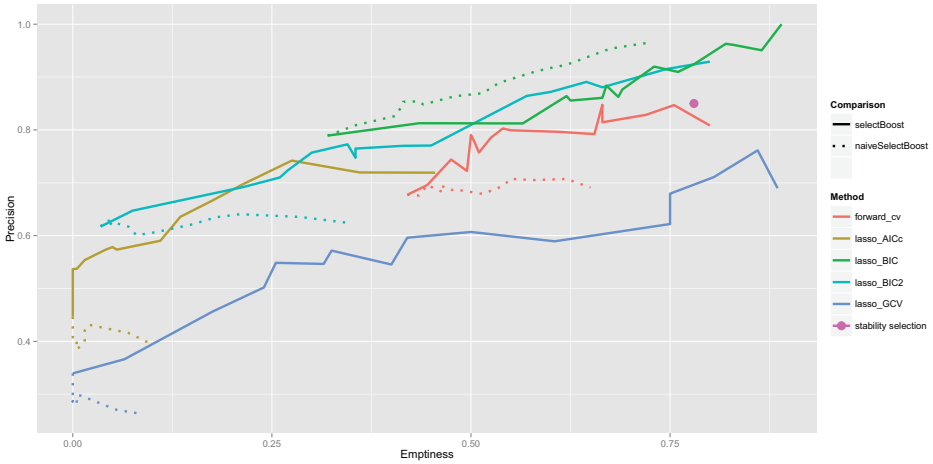


FIGURE 8.3 – *Precision in function of emptiness for all tested method for Situation 1. The selectBoost algorithm is compared to both stability selection and the naiveSelectBoost algorithm.*

Boost algorithm, stability selection does not allow to choose a convenient precision-emptiness trade-off.

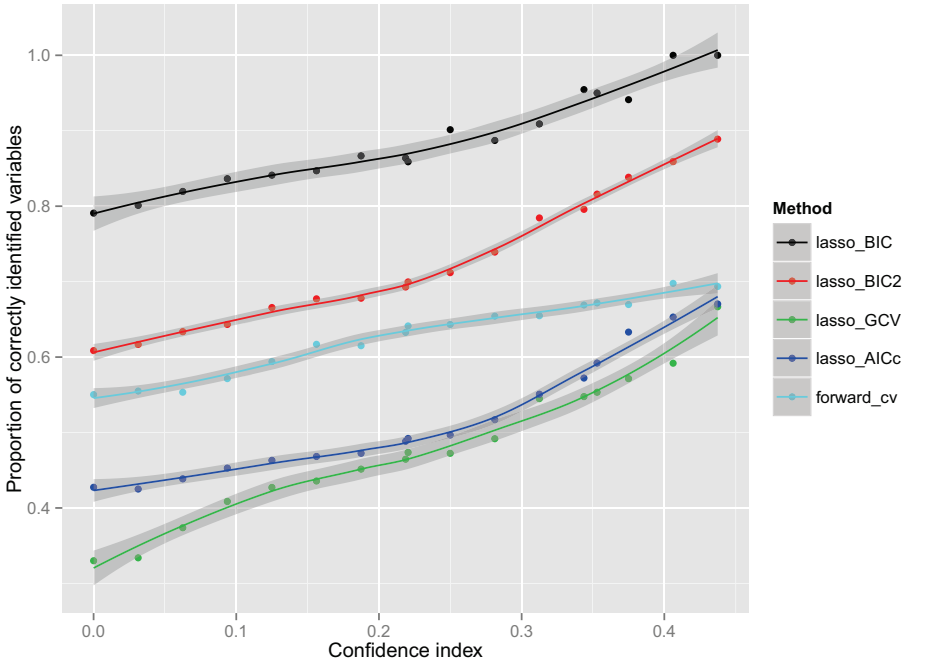


FIGURE 8.4 – *The proportion of correctly identified variables is plotted in function of the confidence index defined in Equation (8.11). The proportion of correctly identified variables is calculated for all variable with a confidence index greater than those mentioned in abscissa. As expected, the greater the confidence index, the higher the proportion of correctly identified variables.*

In the previous section, we mentioned the possibility of using selectBoost to obtain a confidence index, corresponding to one minus the lowest c_0 for which a variable is selected. For each situation, we plot the proportion of correctly identified variables in function of the confidence index (Figure 8.4 for Situation 1 and Supplemental Figures for the others). As expected, the

proportion of correctly identified variables increases with the increase of the confidence index ; it is interesting to note that this increase seems to be linear and that the slope of this linear increase seems not depend on the initial variable selection method *select* which is used.

8.5 Application to the diabetes dataset

We decided to apply our algorithm to the diabetes dataset used by Efron *et al.* (Efron *et al.* 2004). This dataset contains 10 variables which are age, sex, body mass index, average blood pressure and six serum measurements and a quantitative response of interest that is a measure of the evolution of the diabetes disease one year after baseline. As proposed, we included interaction terms, resulting in a 64 explanatory variables dataset.

We first applied the Lasso to this dataset (see Figure 8.5 left for the whole path of the solution). We used cross-validation to choose the appropriate level of penalization (*i.e.* the λ parameter in Equation 8.6). This results in a selection of 22 variables.

We then applied our selectBoost algorithm on the Lasso with penalty parameter chosen by cross-validation. We used a wide range of the c_0 parameter, starting from 1 to 0.35 by step of 0.05 (see Figure 8.5 right). For each step, the probability of being included in the support was calculated with 500 simulations as described in the Algorithm 1. We set the threshold to 0.95 to avoid numerical instability. We used our algorithm with the Lasso and selected the regularization parameter with the AICc.

As previously mentioned, when $c_0 = 1$ our algorithm is equivalent to the Lasso and thus ends with a selection of 22 variables. At the opposite, when using the maximal x_0, our algorithm ends with a selection of only two variables : the body mass index and the average blood pressure. The interesting point is that the these two variables are neither the two first covariables selected by the Lasso or the two variables with the highest coefficients (see Figure 8.5 left). This demonstrates that our algorithm can be very useful to determine which are the variables that are selected with confidence and that it does not simply result in the choice of the variable with the highest coefficients.

8.6 Discussion

We introduced the selectBoost algorithm which uses intensive computation to select variables with high precision. The user has the choice between using this algorithm to produce an confidence index or choosing an appropriate precision-emptiness trade-off to select variables with confidence. The main idea of our algorithm is to take into account the correlation structure of the data and thus use intensive computation to select the reliable variables. We prove the performance of our algorithm through simulation studies in various settings. Indeed, we succeed in improving the precision of all tested initial variable selection methods with a relative stability on

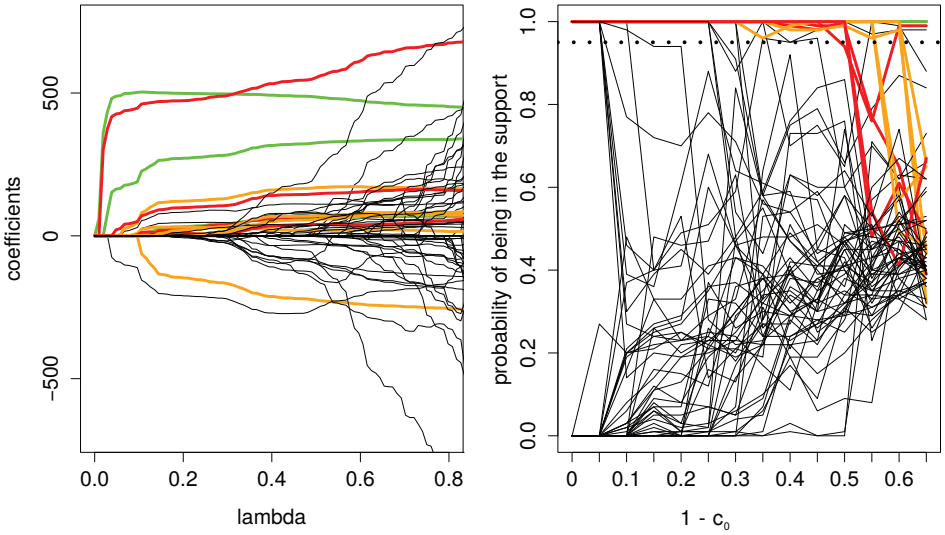


FIGURE 8.5 – *Colors : the green is for the most reliable variables selected by the selectBoost algorithm (confidence index of 0.65 ; orange is for intermediate confidence index (0.55) and red for low confidence (0.45)). Left : evolution of the coefficients in the lasso regression when the sparsity parameter lambda is varying. Right : evolution of the probability of being in the support of the regression when the confidence index is varying. The dotted line represents the threshold of 0.95. The confidence index is calculated as the abscissa at which the probability of being in the support of a variable goes for the first time bellow this threshold.*

recall and Fscore. Our results open the perspective of a precision-emptiness trade-off which may be very useful in some situations where many regressions have to be made (in network reverse-engineering in which we have a regression per vertex). In such a context our algorithm may be used in an experimental design approach. The application to a real dataset allow us to show that the most reliable variable are not necessarily those with the highest coefficient. The selectBoost algorithm is a powerful tool that can be used in every situation where reliable and robust variable selection has to be made.

Troisième partie

Perspectives et conclusions

9 Vers une modulation orientée du programme génique

Les éléments de ce chapitre n'ont pas encore été soumis à publication. Dans ce chapitre, nous utiliserons les concepts et les méthodologies développées dans les chapitres précédents pour déterminer des cibles d'intervention dans le programme génique. Ces cibles seront choisies afin qu'elles orientent le programme génique de l'état tumoral vers l'état sain. Au moment de la rédaction de ces lignes, les validations biologiques des prédictions mathématiques sont en cours. Cependant, les résultats préliminaires semblent encourageants.

9.1 Introduction

Le but de ce chapitre est de donner la méthodologie que nous avons employée afin de déterminer des cibles d'intervention (voir Définition 46 dans ce chapitre). Avant de poursuivre, précisons notre but. Comme nous l'avons dit dans la partie introductive, notre but final est de trouver les bonnes cibles d'intervention afin de reprogrammer un programme génique tumoral en un programme génique sain. Avant de parvenir à cette fin, nous nous sommes donnés un objectif intermédiaire. Nous proposons une modélisation dans laquelle les gènes sont considérés comme étant soit une cible potentielle soit un marqueur. Nous définissons alors :

Définition 46 (Cible potentielle) *Une cible potentielle est un gène différentiellement exprimé aux temps précoces (60 ou 90 minutes).*

Définition 47 (Marqueur) *Un marqueur est un gène différentiellement exprimé et qui n'est pas une cible potentielle.*

Nous décidons alors de retenir l'ensemble des gènes qui sont des marqueurs positivement différentiellement exprimés chez les patients ayant un état cancéreux mais qui ne sont pas différentiellement exprimés chez

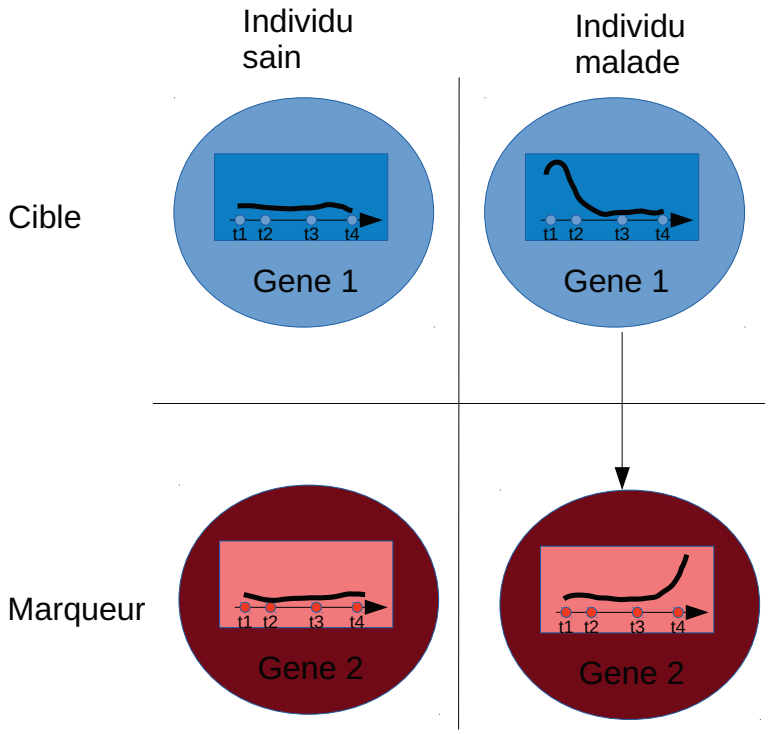


FIGURE 9.1 – *Nous cherchons des couples de gènes qui ne sont pas exprimés chez les individus sains mais qui forment un couple cible marqueur chez les individus malades. De plus, la cible de ce couple doit être un régulateur important pour le marqueur du couple dans le réseau des individus malades. L'idée est ensuite de faire une expérience d'inhibition du gène cible 1 chez les individus malades, en espérant une réduction significative du gène marqueur 2 chez ces mêmes individus. Si cela se produit, et si aucune modification non prévue dans les expressions des autres gènes est observée, nous serons parvenus à moduler le système malade vers le système sain.*

les patients sains. L'objectif que nous nous fixons est alors de déterminer les cibles des patients manifestant un état cancéreux, qui, une fois la cible d'expérience d'intervention les inhibant, réduiront les expressions des marqueurs ainsi choisis. La Figure 9.1 résume nos choix.

Nous sommes partis du jeu de données analysé tout au long de cette thèse. Mais à partir de ce même jeu de données, nous avons aboutis à six listes de couples cibles-marqueurs différentes. Pour cela nous avons fait varier la normalisation du jeu de données ainsi que la méthode d'inférence (voir Figure 9.2). Le choix final des couples cibles-marqueurs réside dans l'intersection des résultats des six listes. Cela permet de gagner en robustesse, et ainsi, nous maximisons les chances de réussite de notre procédure de modulation orientée.

Pour inférer le réseau nous avons utilisé une version évoluée de Cascade nommée Patterns. Nous détaillons les changements dans la section suivante.

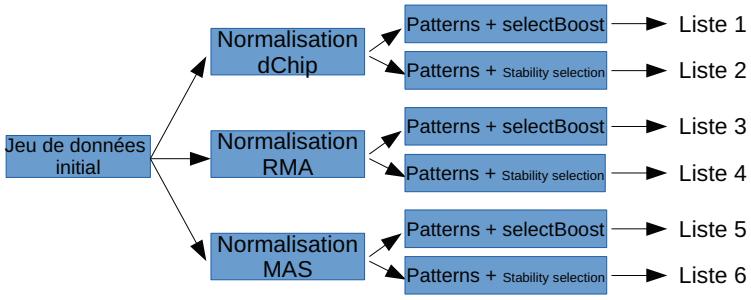


FIGURE 9.2 – *Méthode d'obtention des six listes.*

9.2 De Cascade vers Patterns

La modélisation des réseaux en cascade proposée par la librairie d'extension Cascade [1] souffre de plusieurs limitations. C'est dans le but de lever ces limitations et de généraliser les possibilités d'application que nous avons fait évoluer Cascade vers Patterns [2]. Nous gardons le même modèle linéaire, tel que défini par l'équation 5.2. Dans ce modèle, une importance première est donnée aux matrices inconnues $\boldsymbol{F}$ et $\boldsymbol{\Omega}$. Les matrices $\boldsymbol{F}$ permettent de déterminer *comment* un gène agit sur un autre gène tandis que la matrice $\boldsymbol{\Omega}$ permet de déterminer la *puissance* du lien entre deux gènes (et en particulier, s'il existe ou non un lien). Nous rappelons également l'importance de la classification en patterns temporels, laquelle est utilisée dans l'inférence de réseau. La librairie permet alors une extension de ces trois éléments clefs :

- **pour la classification en groupes temporels** : dans la librairie Cascade, nous avons imposé la forme et le nombre des patterns temporels. Dans Patterns, il devient possible de constituer une classification avec un nombre de groupes choisis et sans contrainte sur les patterns recherchés ;
- **pour les matrices $\boldsymbol{F}$** : dans la librairie Cascade, la forme de ces matrices est contrainte (voir équation 5.3). Dans Patterns, c'est à l'utilisateur de choisir les contraintes (ou de n'en proposer aucune) ; par ailleurs, il est également possible de choisir si un groupe de gènes peut ou non interagir avec un autre groupe de gènes en imposant à certaines matrices $\boldsymbol{F}$ d'être nulles ;
- **pour la matrice $\boldsymbol{\Omega}$** : dans la librairie Cascade, la seule méthode proposée pour l'estimation des paramètres est le Lasso. Dans Patterns, le choix de la méthode de sélection de variables est laissée au choix de l'utilisateur.

9.3 Le choix de la normalisation

La normalisation des puces à ADN est un sujet que nous n'avons pas encore abordé dans cette thèse. La nécessité de normalisation des puces à ADN vient du fait que deux puces à ADN ne sont pas directement comparables entre elles. Par exemple, certaines peuvent être globalement

1. Disponible librement ici : `http://www-math.u-strasbg.fr/genpred`
2. Disponible librement ici : `http://www-math.u-strasbg.fr/genpred`

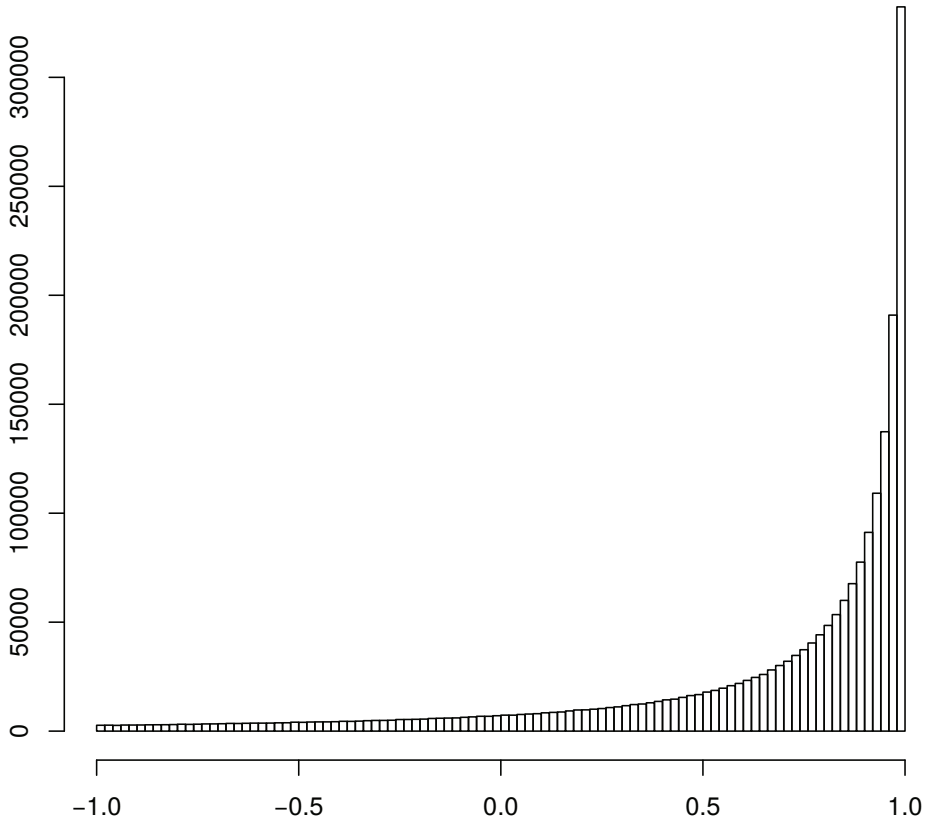


FIGURE 9.3 – *Histogramme des corrélations entre les expressions de gènes obtenues par les méthodes de normalisation dChip et RMA*

plus lumineuses que d'autres, conduisant à des expressions d'ARNm globalement plus élevées. Pour pallier ce problème, il faut normaliser les données issues de puces à ADN (Quackenbush 2002, Wu et Irizarry 2004, Järvinen *et al.* 2004, Cope *et al.* 2004). Il n'existe pas de normalisation de référence. Pour trouver les cibles d'intervention, nous avons sélectionné celles qui étaient robustes à un changement de normalisation. Nous avons choisi d'utiliser trois normalisations différentes : RMA (Irizarry *et al.* 2003), MAS (Wu et Irizarry 2004), et dChip (Li et Wong 2003). Une étude de la corrélation linéaire entre les expressions de gènes pour les différentes méthodes montre une bonne cohérence (voir, par exemple Figure 9.3).

9.4 Le choix de la méthode de sélection

Tout comme le choix de plusieurs normalisations nous conduit à obtenir des cibles plus fiables (puisque ne dépendant pas de ladite normalisation), nous allons utiliser deux algorithmes pour la sélection de variables pour l'inférence de la matrice $\mathbf{\Omega}$:

— nous utiliserons l'algorithme Lasso (Tibshirani 1996), avec le critère BIC pour la sélection du paramètre de pénalité ; cet algorithme sera amélioré par l'utilisation de l'algorithme selectBoost, tel que présenté dans le Chapitre 8,

— nous utiliserons l'algorithme "stability selection" (Meinshausen et Bühlmann 2010).

9.5 Procédure suivie

Nous avons alors, pour chaque couple de méthode de sélection, de normalisation, suivi les étapes suivantes (nous avons également distingué les patients ayant la forme indolente et ceux ayant la forme agressive de la maladie) :

1. Déterminer les cibles potentielles et les marqueurs.
2. Faire une classification non-supervisée sur les cibles potentielles.
3. Faire une classification non-supervisée sur les marqueurs.
4. Inférer un réseau de gènes avec l'ensemble des groupes des étapes 2 et 3, en imposant les contraintes suivantes sur les matrices $\boldsymbol{F}$:

   — elles ont la forme définie par l'équation 5.3,
   — les seules interactions possibles sont celles qui partent d'un gène cible potentielle vers un gène marqueur.
   — sélectionner les hubs parmi les cibles potentielles : ce sont eux qui auront le plus d'impact sur nos marqueurs.

9.6 Résultats obtenus

Ce travail étant en cours de réalisation (et c'est pourquoi il est placé dans la partie "Conclusions et perspectives") peu de résultats ont été obtenus. Cependant, la partie statistique de ce travail peut être considérée comme étant terminée ; les résultats obtenus doivent maintenant être analysés biologiquement. Toutefois, un élément nous semble relever de la première importance. En effet, pour chaque état (cancéreux en stade agressif ou indolent), nous avons réalisé notre procédure six fois en changeant la normalisation des micropuces (trois normalisations distinctes) et en changeant la méthode d'inférence (deux méthodes). Malgré tout, certaines cibles désignées par notre procédure sont communes à l'ensemble des résultats, ce qui montre la robustesse de notre procédure et nous laisse espérer des résultats biologiques intéressants.

Conclusions 10

Dans ce chapitre, nous allons établir un bilan de nos travaux avant de proposer quelques perspectives.

10.1 Résultats

Nos travaux ont conduit à l'obtention de résultats intéressants, tant d'un point de vue biologique que statistique. Il est impossible de distinguer les deux, tant les résultats se répondent et s'imbriquent. Cette thèse propose un état de l'art sur l'inférence de réseau par ingénierie inverse et sur la sélection de variables dans la régression linéaire. Nous avons porté un intérêt tout particulier à la régression Lasso.

Nous avons ensuite proposé une modélisation statistique pertinente au modèle biologique proposé. C'est ainsi que nous avons développé la notion de réseau en cascade. Cette modélisation statistique nous a permis dans un premier temps de reconstituer les différents réseaux de gènes suite à la stimulation du BCR dans les lymphocytes B chez des patients atteints de leucémie lymphoïde chronique et chez des sujets sains. Cette inférence nous a révélé quels étaient les gènes les plus importants. Cette étape, d'essence purement descriptive, permet une première compréhension (Vallat *et al.* 2013). Par ailleurs, la performance de notre méthode statistique a été démontrée *in silico* par comparaison avec des méthodes de référence.

Le programme génique soutenant la prolifération cancéreuse est, comme nous l'avons vu au Chapitre 1, un système complexe. L'inférence du réseau sous-jacent est une manière de représenter la structure de ce programme génique. En utilisant une méthode basée sur des régressions pénalisées, nous avons également pu en capturer la dynamique. Mais une meilleure compréhension d'un système complexe passe forcément par la capacité de prédire son comportement suite à une altération de son état. C'est pourquoi nous avons inhibé l'expression d'un des gènes majeurs, DUSP1, pour les patients atteints d'un cancer. Nous avons alors utilisé notre modèle statistique pour établir des prédictions, que nous avons comparées aux observations faisant suite à l'expérience biologique. Il est apparu que notre modèle est capable, pour peu que nous considérions des temps proches du moment de l'intervention, de prédire avec succès l'impact sur l'expression des autres gènes de l'inhibition de DUSP1 (Vallat *et al.* 2013).

Nous avons ensuite porté nos efforts pour développer et mettre sous forme de librairie R la modélisation proposée dans Vallat *et al.* (2013). En particulier, nos efforts ont été les suivants (Jung *et al.* 2014) :

— utilisation de la propriété d'invariance d'échelle pour la détermination d'un seuillage optimal pour les liens dans le réseau,
— développement d'une méthodologie pour simuler des réseaux avec une topologie en cascade, en se basant sur le principe d'attachement préférentiel,
— utilisation de notre méthodologie pour retrouver et améliorer des résultats déjà publiés ; d'une part, sur notre jeu de données initial (voir Annexe B) et d'autre part, sur un jeu de données pris dans la littérature (Annexe C),
— visualisations intuitives de la dynamique du réseau.

Nous avons alors proposé un algorithme permettant d'améliorer la précision des méthodes de sélection de variables, dans le cadre de la régression linéaire (Chapitre 8). Notre algorithme permet soit de choisir un compromis entre proportion de modèle vide (c'est-à-dire, un modèle dans lequel aucune variable ne peut être sélectionnée avec confiance), soit de proposer un indicateur de confiance qui permet de ranger les variables en fonction de la confiance que nous avons à les inclure dans le support de la régression. Nous avons comparé notre algorithme avec un algorithme de référence et avons démontré l'intérêt de notre algorithme sur un exemple réel.

10.2 Perspectives de travail

Suite à ces travaux, plusieurs perspectives peuvent être envisagées. Tout d'abord, nous avons présenté dans le Chapitre 9 une perspective de travail sur laquelle des travaux sont déjà en cours. En effet, nous y avons proposé une méthode permettant de sélectionner les cibles optimales sur lesquelles notre modélisation mathématique prédit un impact maximal "dans le sens de la guérison". Cette dernière notion est basée sur l'expression des gènes qui s'expriment aux temps tardifs ; notre but est alors de cibler des gènes aux temps précoces dont l'inhibition entraînera l'inhibition de gènes aux temps tardifs différentiellement exprimés chez les patients cancéreux et non différentiellement exprimés chez les sujets sains. Pour trouver ces cibles optimales nous avons essayé de proposer la méthode la plus robuste possible. En particulier, nous avons tenu compte des points suivants :

— utilisation de trois méthodes de normalisation des micropuces,
— utilisation de deux méthodes de sélection de variables robustes (dont celle présentée au Chapitre 8),
— évolutions méthodologiques du modèle d'inférence de réseau (voir Chapitre 9).

Mais d'autres perspectives sont envisageables suite à ce travail. En particulier, l'idée de planification d'expérience semble être une perspective naturelle, et notre méthode permet de choisir un compromis entre modèle vide et précision. Il devient alors possible de se fixer un niveau de précision,

et de faire des expériences d'interventions successives pour réduire le nombre de modèles vides (dans ce cas il s'agirait de gènes sans régulateurs) tout en gardant le même niveau de précision. De ce point de vue, il serait intéressant de comparer notre approche avec celle développée dans Rau *et al.* (2010) où la méthode ABC permet également de connaître les sous-réseaux du réseau total inféré avec peu de certitude.

Une autre perspective intéressante serait de développer le travail qui a été soumis lors des journées SFdS (voir Annexe F). Si nous posons l'hypothèse que les différents réseaux (sains et tumoraux) partagent une partie de leur topologie, il devient possible d'utiliser cette information commune pour améliorer l'inférence. En particulier, cela nous permettrait également de déterminer de façon fiable le "cœur" du réseau commun à tous les états des patients et des sujets sains.

10.3 Conclusion

Il est temps de mettre un point final au manuscrit de cette thèse. Il y a eu dans cette thèse plusieurs défis que j'ai du relever.

De formation purement mathématicienne, j'ai du me plonger dans l'univers de la biologie. Tout le but de ma démarche a été d'apporter aux biologistes les meilleures modélisations utilisant les meilleurs outils statistiques possibles. Pour réussir à faire cela, j'ai été contraint de saisir toutes les subtilités du système complexe que nous avons étudié. Il serait faux de dire que cet apprentissage a été facile, d'autant plus que le dialogue avec les biologistes est un exercice délicat. De la biologie aux mathématiques, les raisonnements ne sont pas toujours les mêmes, et les mots peuvent revêtir des significations équivoques.

Mais au final, ces difficultés furent surmontées, laissant ainsi la place à une collaboration heureuse. En effet, outre les articles déjà publiés et présentés dans cette thèse, l'article en cours de publication qui fait l'objet du Chapitre 8, se dresse la perspective que nous avons évoqué dans le Chapitre 9. De plus, du fait d'appartenir à une équipe de biologie, j'ai pu participer à plusieurs projets d'importance qui n'ont été évoqués que rapidement dans la préface de ce manuscrit, mais qui devraient conduire très prochainement à des publications.

Quatrième partie

Annexes

Supporting information for Chapter 3

A

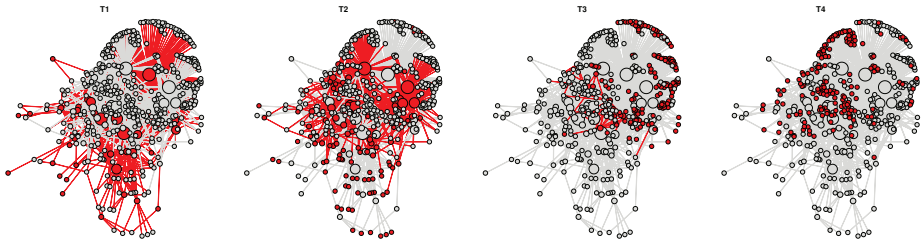
Method	Possible prediction	Designed for large networks	Short description
TD-ARACNE Zoppoli *et al.* (2010)	No	No	Time delay regression combined to ARACNE Margolin *et al.* (2006a) method. ARACNE is an information theory model, based on mutual information.
GeneNet Schafer et Strimmer (2005)	No	Yes	Graphical Gaussian Model (GGM) using partial correlation.
GeneReg Huang *et al.* (2010)	Yes	No	Regression with spline interpolation to increase the number of time points.
Morrissey et al. Morrissey *et al.* (2011)	Yes	No	Dynamic Bayesian Network (DBN).

Table A.1 – *Short description of selected methods used for inference methods comparisons.*

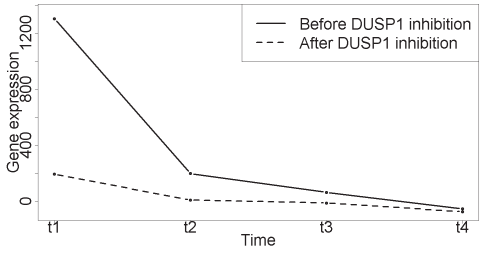
Method	User fixed parameter	Note
TD-ARACNE Zoppoli *et al.* (2010)	The number N of bin in the discretization process	The best performing value of N was retained after sequential tests (N value varying from 6 to 20, by 0.5).
GeneNet Schafer et Strimmer (2005)	None	None
GeneReg Huang *et al.* (2010)	Number of interpolated time points	The ratio (number of interpolated time points) / (number of initial time points) used by the authors has been conserved.
Morrissey et al. Morrissey *et al.* (2011)	None	None

TABLE A.2 – *Settings of selected methods used for inference methods comparisons.*

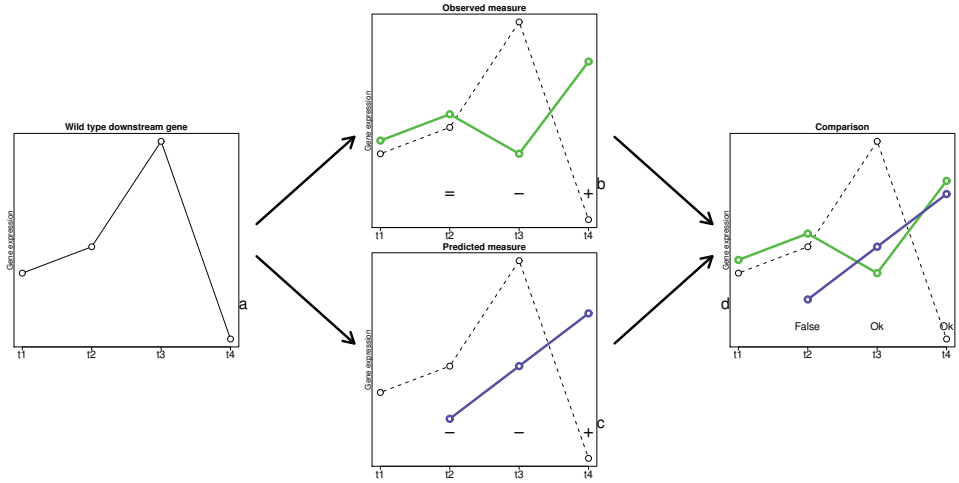
Supplemental Figure A.1 – *Venn Diagram : distribution of the 960 probe sets between the 3 cell groups. A total of 960 probe sets was retained for all the subjects across the three different cell groups. A core of 183 probe sets is shared by the 3 groups.*



Supplemental Figure A.2 – *Each graphic represents genes in a specific categorical time label (1 to 4, from left to right) and their connections, showing how the signal is spreading through the aggressive network.*



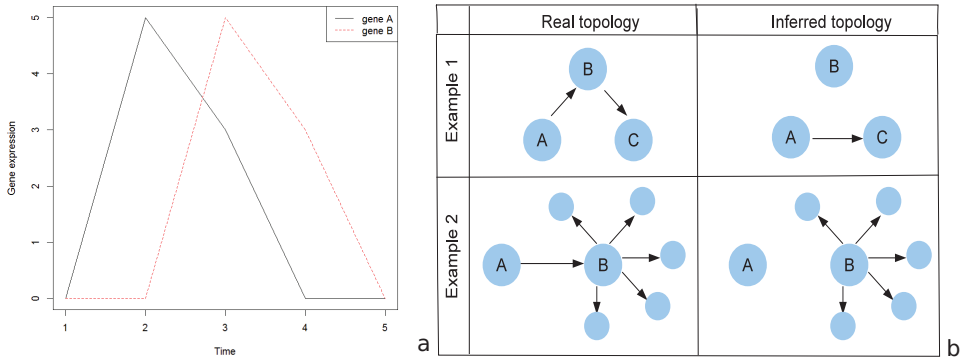
Supplemental Figure A.3 – *DUSP1 is the targeted gene for the knock-down experiment. We show its expression before and after the inhibition experiment.*



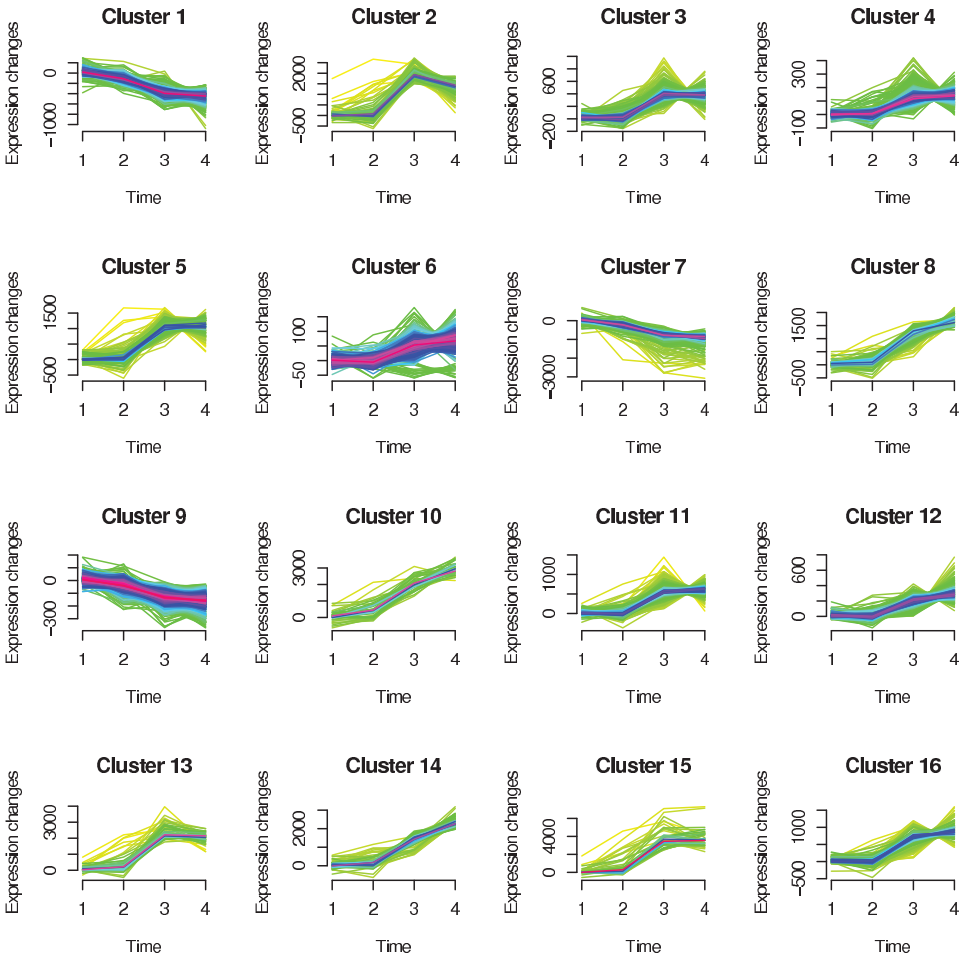
Supplemental Figure A.4 – *Principle of the validation experiment. Graphic (a) represents a gene expression before inhibition of a targeted gene. Graphic (b) shows how this gene expression evolves after silencing this targeted gene whereas graphic (c) shows the predicted gene expression. For these two last graphics, for time t_2 to t_4 we assigned a +,- or = label if gene expression after silencing is respectively greater, smaller, or equal to gene expression before silencing. For this gene, graphic (d) shows that we made two good predictions out of three in this example.*

	Our method	TD-ARACNE Zoppoli *et al.* (2010)	GeneNet Schafer et Strimmer (2005)	GeneReg Huang *et al.* (2010)
Our method	1528	28	39	7
TD-ARACNE Zoppoli *et al.* (2010)	28	5236	87	66
GeneNet Schafer et Strimmer (2005)	39	87	2241	86
GeneReg Huang *et al.* (2010)	7	66	86	1567

TABLE A.3 – *Supplemental Table 3. A comparison of our method with three other benchmark methods applied to the more aggressive CLL data set. Total number of inferred links for each selected method and intersection between the methods, in total common inferred links. The biological data set of patients with more aggressive CLL type is used.*



Supplemental Figure A.5 – *Schematic representation of specific constraints related to prediction abilities in model inference. This ability to predict the transcriptional effect of a modulation in the network is crucial in order to predict a gene expression level modification after a knock-down experiment. For instance, given a situation where a gene A regulates the expression of a gene B (with a time lag between activation of gene A and gene B as schematically shown in (a)), which in turn regulates gene C, we want to predict the absence of link between B and C if gene A is knocked-down. Importantly, this predictive capacity requires much more complexity than inference alone. More than inferring a network topology, a predictive method should be able to learn how the biological signal spreads in this network. To go further, the best algorithms for reverse-engineering are not necessarily the best methods for predicting purposes, as explained in (b) with two simple examples. In the first example, a real network is composed of a gene A that activates a gene B, which in turn activates gene C (upper-left quadrant). An inference method could infer a statistical link between A and C, leading to two false negative links (two existing links are not presents) and one false positive link (upper-right quadrant). However, to predict gene C's expression, given the expression of gene A, this inference method will probably give adequate results. In the 2nd example (lower-right quadrant), a better inference method could give six true positive inferred links and only one false negative, omitting the link between A and B. However, in this case, we have a dramatic situation for prediction purpose as gene A cannot activate gene B anymore.*



Supplemental Figure A.6 – *Significance of selected patterns in the clustering step. To evaluate the relevance of our selected patterns used for enrichment, we compared these patterns with various temporal gene clusters obtained with a gold standard unsupervised clustering method. One of the most widely used clustering methods is fuzzy c-means . The preponderant aspect of this algorithm relies on the fuzzy parameter that allows taking into account the inherent noise of transcriptional data (when this parameter increases, more genes are randomly assigned into clusters). For comparison purposes, we focused on the biological data set of patients with more aggressive CLL type and we first select relevant genes with Limma , using a p-value of 0.01. An unsupervised temporal clustering of the 8,113 genes retained with Limma is then performed showing 16 distinct clusters. Importantly, these clusters emphasize the existence of genes with transient expressions (peaks) at t1 (cluster#1, 7, 9), at t2 (within cluster#2, 4), at t3 (cluster#2, 3, 4, 11, 13, 15) and t4 (cluster#5, 6, 8, 10, 12, 14, 16) ; as shown by our method. The fact that through this unsupervised clustering method we reach similar patterns that those produced by our method, confirms the pertinence of our own gene selection process.*

Application 1 : B

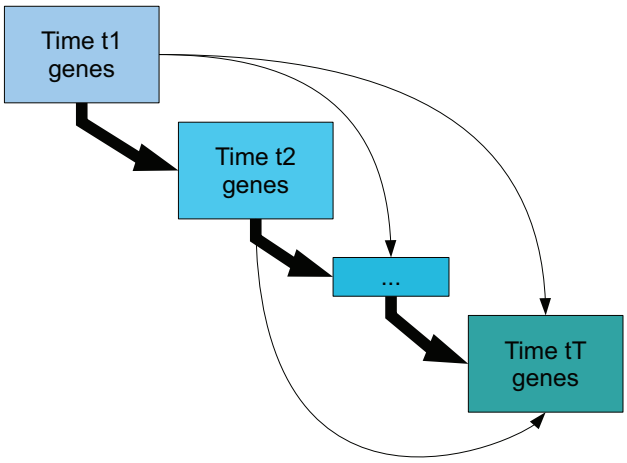
B.1 Introduction

In a cell, after a specific activation, a gene contained in the DNA can be expressed as RNA molecules that are later traduced in proteins that will sustain the cell response (Crick *et al.* (1970)). Cells are in continuous contact with their environment within the organism and display an adapted response to its modifications (Barabási et Oltvai (2004)). For this, each transient environmental modification activates cell' surface receptors (and co-receptors) that induce multiple integrated signaling cascades whose ultimate events are expression of specific transcriptional factors (TF). These first TF induce the expression of other genes within the cell. Some of these genes code themselves for TF or transcriptional regulators (TR) that induce sequential activation of other genes. At the end, concerted expression of these multiple genes induces protein expressions that are the substratum of the adapted cellular reaction to the initial stimulus.

One Common tool to analyze such complex systems is regulatory networks (RN). When studying transcriptional data, this RN is called a gene regulatory network (GRN) in which the vertex represent genes and edges represent potential (orientated) interactions between these genes.

Since the emergence of high-throughput technologies that allow simultaneously measuring mRNA expression of thousands of genes, many tools have been developed to analyze and reverse engineer their underlying GRN (Hecker *et al.* (2009), Bar-Joseph *et al.* (2012)). These methods should be splitted between static and time dependent methods. While the former relies on the assumption than co-expressed genes share some biological characteristics, the latter infers a directed network. In this last case, another important distinction should be made between temporal phenomenom induced by exogenous stimulus (e.g, stress response) or endogenous stimulus (*e.g.*, cell cycle) (Zhu *et al.* (2007), Luscombe *et al.* (2004), Yosef et Regev (2011)). These two stimulii result in different network topologies. Indeed, after an exogenous stimulus, networks topologies seem to have larger hubs and shorter paths leading to a quick response to external conditions (Luscombe *et al.* (2004)) and resulting in a cascade topology (Figure 1).

The Cascade package is a tool dedicated to the analysis of microarray data and to the inference cascade networks. The statistical tools provided in this library are based on the methodology described by Vallat *et al.* (Vallat *et al.* (2013)) and contained several major improvements described here.



Supplemental Figure B.1 – *Cascade networks are temporal nested networks*

B.2 Installation requirements

Following software is required to run the Cascade package :

— R (> 2.14.2). For installation of R, refer to http ://www.r-project.org.

— R-packages : `abind`; `animation`; `cluster`; `datasets`; `graphics`; `grDevices`; `igraph`; `lars`; `lattice`; `cccccc`*; `magic`; `methods`; `nnls`; `splines`; `stats`; `stats4`; `survival`*; `tnet`; `utils`; `VGAM`.

To install them :

— without stars :

```
> install.packages("name_of_the_package")
```

— with one star :

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("name_of_the_package")
```

Once the *Cascade* package is installed, you can load the package by :

```
> library(Cascade)
```

B.3 Data pre-processing

To illustrate our approach we will analyze a microarray dataset of the transcriptional response of healthy B-cells after B-cell receptor stimulation (Vallat *et al.* (2007)). Our dataset (part of GSE39411, (Vallat *et al.* (2007))) is separated in two files : the first, `micro_S`, corresponds to the stimulated gene expressions while the second, `micro_US`, corresponds to the unstimulated gene expressions. In other words, `micro_US` is the control dataset. You can load these data by :

```
> data(micro_S)
> data(micro_US)
```

Each of the these dataset corresponds to 54613 genes measured through 4 time points and 6 subjects (we have repeated longitudinal data).
These data need to be coerced into a `micro_array` class. The matrix with the microarray measurements has to be of size $N \times K$ where N is the number of genes and $K = T \times P$ where T stands for the number of time points and P for the number of subjects. The first T columns are the gene expressions for subject 1, the following T are the gene expressions for subject 2... In our case :

```
> colnames(micro_S)
```

```
 [1] "N1_S_T60"  "N1_S_T90"  "N1_S_T210" "N1_S_T390"
 [5] "N2_S_T60"  "N2_S_T90"  "N2_S_T210" "N2_S_T390"
 [9] "N3_S_T60"  "N3_S_T90"  "N3_S_T210" "N3_S_T390"
[13] "N4_S_T60"  "N4_S_T90"  "N4_S_T210" "N4_S_T390"
[17] "N5_S_T60"  "N5_S_T90"  "N5_S_T210" "N5_S_T390"
[21] "N6_S_T60"  "N6_S_T90"  "N6_S_T210" "N6_S_T390"
```

To coerce the data toward a `micro_array` class, you may just use the `as.micro_array` function :

```
> micro_S<-as.micro_array(micro_S,time=c(60,90,210,390),subject=6)
> micro_US<-as.micro_array(micro_US,time=c(60,90,210,390),subject=6)
```

In addition of the matrix of microarray measurements, this class also contains the name of genes, their group, the first time at which they are expressed, the time points at which they are measured, and the number of subjects. Primarily, method `print` summarizes these informations :

```
> print(micro_S)

This is a micro_array S4 class. It contains :
 - (@microarray) a matrix of dimension  54613 * 24
         .... [gene expressions]
 - (@name) a vector of length  54613  .... [gene names]
 - (@group) a vector of length  1  .... [groups for genes]
 - (@start_time) a vector of length  1
         .... [first differential expression for genes]
 - (@time)a vector of length  4  .... [time points]
 - (@subject) an integer  .... [number of subject]
```

While method `print` gives the structure of the object, method `head` gives an overview of the data :

```
> head(micro_S)

The matrix :

          N1_S_T60 N1_S_T90 N1_S_T210
1007_s_at    136.1    116.6     127.6
1053_at       32.0     43.3      31.3
117_at        78.0     63.5      57.9
121_at       201.8    209.2     208.8
1255_g_at     16.3      8.0      15.8
1294_at      196.8    198.7     163.9
...

Vector of names :
[1] "1007_s_at" "1053_at"   "117_at"    "121_at"
[5] "1255_g_at" "1294_at"
...
Vector of group :
[1] 0
...
Vector of starting time :
[1] 0
...
Vector of time :
[1]  60  90 210 390

Number of subject :
[1] 6
```

Entries `Vector of group` and `Vector of starting time` are set to 0 because they are no yet defined. They will be completed automatically when using gene selection functions of this package. Otherwise, it should be completed by the user.

Once the data are coerced into the `micro_array` class, this package allows doing gene selection and reverse-engineering of the network.

B.4 Gene selection

The selection step requires at least two sets of data. The selection function will select genes differentially expressed in one condition compared with the other. If only one experimental condition is provided (e.g., unstimulated control data omitted), it will be compared to a flat and null pattern.

In this package gene selection mainly relies on the R-bioconductor `cccccc` package (Smyth (2005)). The `cccccc` package allows selecting genes that are differentially expressed between two conditions. In our case, these two conditions are "*stimulated*" and "*unstimulated*". The method relies on linear models and on improved bayesian t-tests (Smyth (2005)). Basically, to find the 50 more significant expressed genes you will use :

```
> Selection<-geneSelection(x=micro_S,y=micro_US,
   tot.number=50,data_log=TRUE)
```

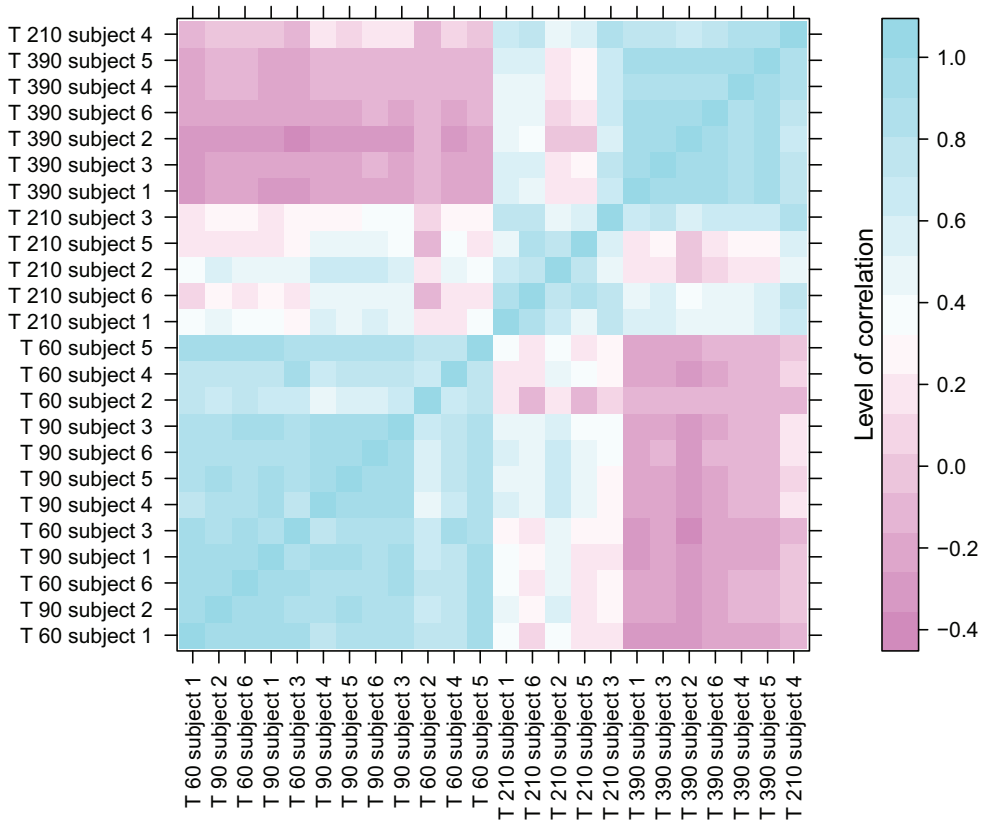
The `data_log` option (default to TRUE) indicates that the data are logged before analysis. This function returns an object of class `micro_array`, with the difference "stimulated" (S) minus "unstimulated" (US) of the 50 more significant expressed genes ; as the `data_log` option is here activated, we get :

$$\log(S) - \log(US) = \log\left(\frac{S}{US}\right).$$

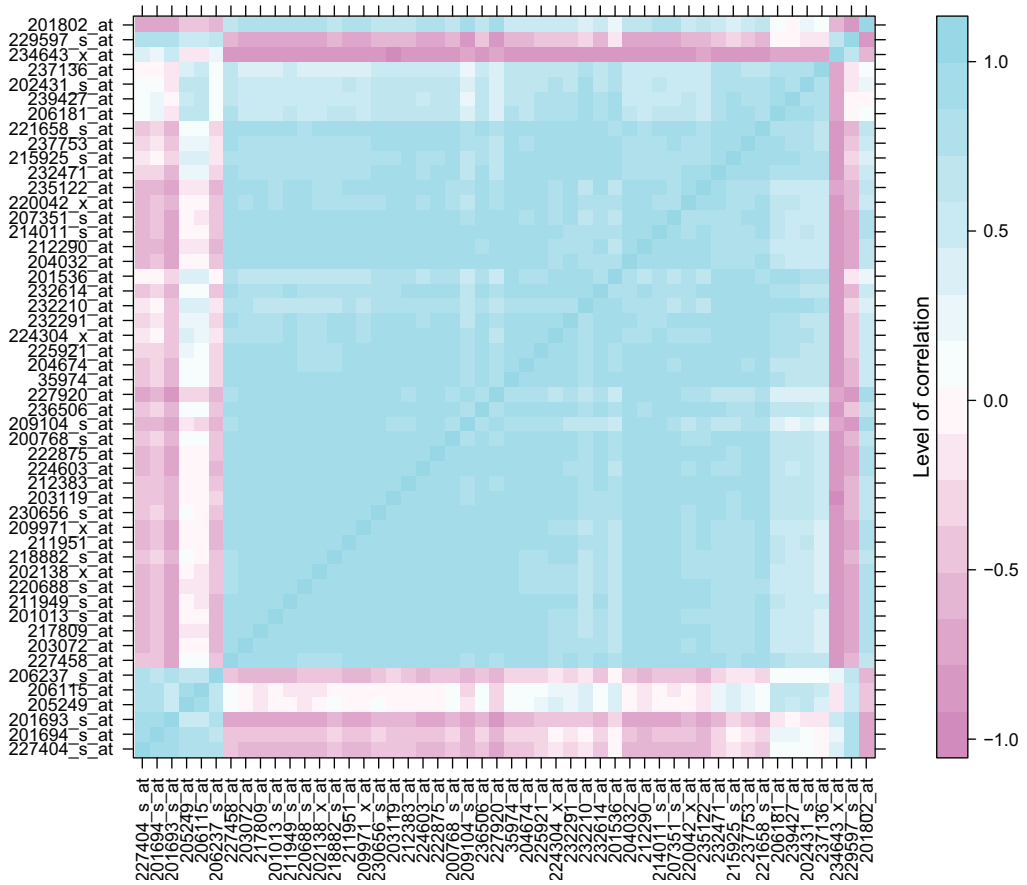
Notice that the `group` and `start_time` are filled out automatically.

Applying the `summary` method prints the structure of Pearson linear correlation for subjects (see Figure B.2) and the structure of Pearson linear correlation for genes (see Figure B.3) :

```
> summary(Selection)
```



Supplemental Figure B.2 – *Correlation between subjects*



Supplemental Figure B.3 – *Correlation between genes*

Note that a hierarchical clustering (function `agnes` of package `cluster`) is performed before plotting the result. This allows pointing out some structures, as correlated objects will be close in the graph.

If we want to select genes that are differentially expressed at specific time points we use the option `wanted.patterns` :

```
> #If we want to select genes that are differentially
> #at time t60 or t90 :
> Selection<-geneSelection(x=micro_S,y=micro_US,tot.number=30,
   wanted.patterns=
   rbind(c(0,1,0,0),c(1,0,0,0),c(1,1,0,0)))
```

You may want forbid some patterns thanks to the `forbidden.patterns` option.

If we wish select genes that have a differential maximum of expression at a specific time point, we may use the `genePeakSelection` method. Basically, this function selects genes that are differentially expressed at desired time point, and which differential expression is significantly higher at this time point :

```
> Selection<-genePeakSelection(x=micro_S,y=micro_US,1,
   abs_val=FALSE,alpha_diff=0.01)
```

If there are more than two microarrays of interest, geneSelection may be used with a list of microarrays as first argument, and a list specifying the contrast as a second argument :

First element : "condition", "condition&time" or "pattern". The "condition" specification is used when the overall goal is to compare two conditions. The "condition&time" specification is used when comparing two conditions at two precise time points. The "pattern" specification allows to choose at which time points selected a gene should be expressed or not.

Second element : a vector of length 2, corresponding to the two conditions that should be compared. If a non-temporal dataset is used as control, it should be the first element of the micro_array list and the option "cont=TRUE" should be used.

Third element : depends on the first element. This element is not needed if "condition" has been specified. If "condition&time" has been specified, then this is a vector containing the time point at which the comparison should be done. If "pattern" has been specified, then this is a vector of 0 and 1 of length T, where T is the number of time points. Time points where differential expression is wanted are provided with 1.

We can now compute an effective selection. As shown in Figure B.4, the early time points (t_1 =60 and t_2 =90) are correlated together and the later time points (t_3 =210 and t_4 =390) are correlated together; this is a fact that is well known in the literature (Yosef et Regev (2011)).

As an illustrating example, the following selection will be used for reverse-engineering :

```
> #Genes with differential expression at t1
> Selection1<-geneSelection(x=micro_S,y=micro_US,
       20,wanted.patterns= rbind(c(1,0,0,0)))
> #Genes with differential expression at t2
> Selection2<-geneSelection(x=micro_S,y=micro_US,
       20,wanted.patterns= rbind(c(0,1,0,0)))
> #Genes with differential expression at t3
> Selection3<-geneSelection(x=micro_S,y=micro_US,
       20,wanted.patterns= rbind(c(0,0,1,0)))
> #Genes with differential expression at t4
> Selection4<-geneSelection(x=micro_S,y=micro_US,
       20,wanted.patterns= rbind(c(0,0,0,1)))
> #Genes with global differential expression
> Selection5<-geneSelection(x=micro_S,y=micro_US,20)
```

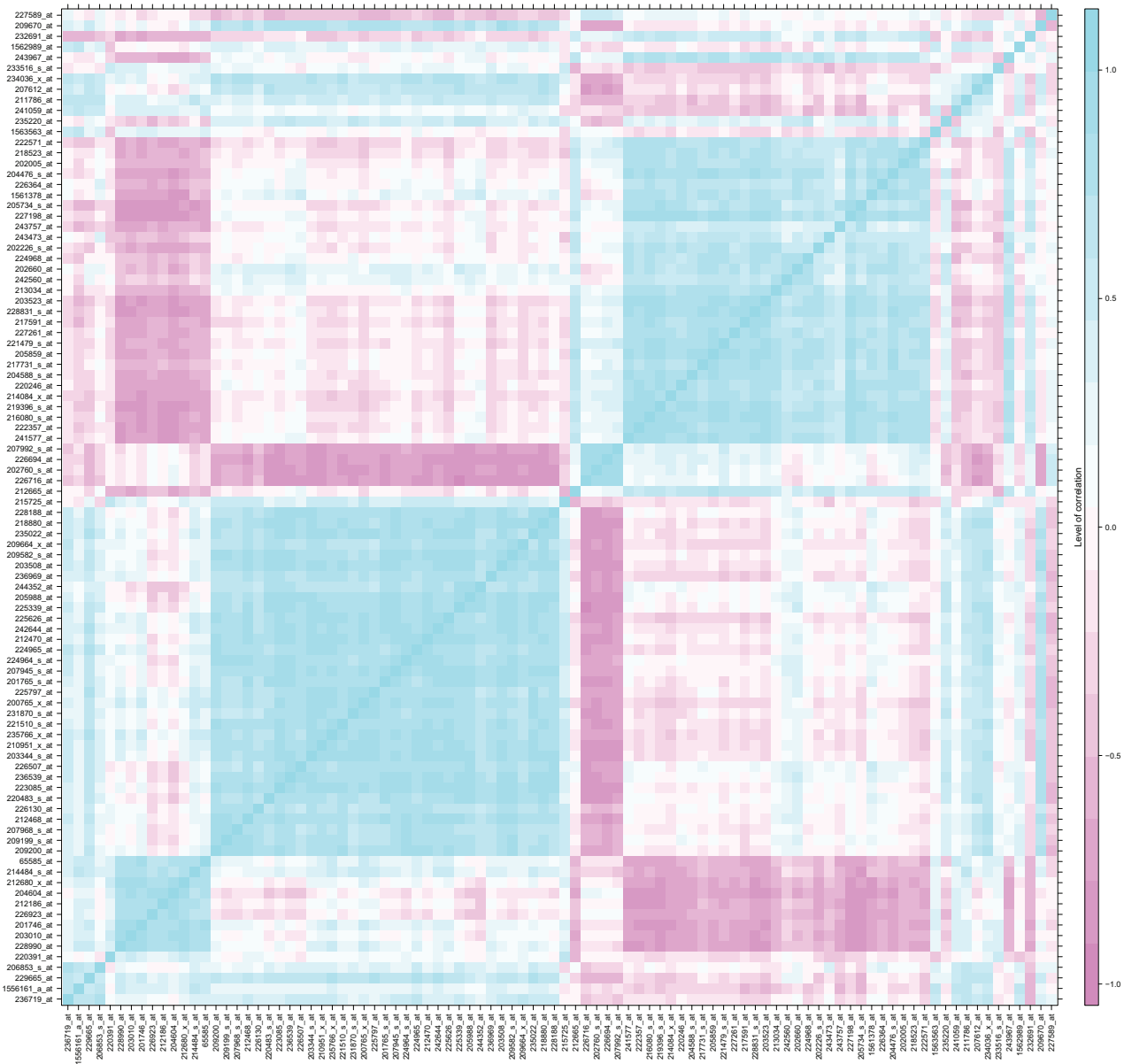
We then make the union between these different selections :

```
> Selection<-unionMicro(list(Selection1,Selection2,
       Selection3,Selection4,Selection5))
> print(Selection)

This is a micro_array S4 class. It contains :
 - (@microarray) a matrix of dimension  74 * 24
         .... [gene expressions]
 - (@name) a vector of length  74  .... [gene names]
 - (@group) a vector of length  74  .... [groups for genes]
 - (@start_time) a vector of length  74
         .... [first differential expression for genes]
 - (@time)a vector of length  4  .... [time points]
 - (@subject) an integer  .... [number of subject]
```

We use the `org.Hs.eg.db` Bioconductor database to match probesets with gene ID :

```
> library(org.Hs.eg.db)
> ff<-function(x){substr(x, 1, nchar(x)-3)}
> ff<-Vectorize(ff)
> #Here is the function to transform the probeset names to gene ID.
>
> library("hgu133plus2.db")
> probe_to_id<-function(n){
   x <- hgu133plus2SYMBOL
   mp<-mappedkeys(x)
   xx <- unlist(as.list(x[mp]))
   genes_all = xx[(n)]
   genes_all[is.na(genes_all)]<-"unknown"
   return(genes_all)
   }
> Selection@name<-probe_to_id(Selection@name)
```

```
> #Prints the correlation graphics Figure 4:
> summary(Selection,3)
```



Supplemental Figure B.4 – *Correlation structure of the final selection*

B.5 Gene regulatory network reverse-engineering

B.5.1 Theoretical background

Our gene regulatory network reverse-engineering method relies on a Lasso penalized estimation of a linear regression model (Tibshirani (1996)). Before describing our model, we make some general reminders of the Lasso estimator.

The Lasso estimate

Suppose that we have data $(\boldsymbol{x_{i.}}, y_i)_{i=1,\cdots,N}$ where the $\boldsymbol{x_{i.}} = (x_{i1}, \cdots, x_{ip})^T$ are the predictors while the y_i are the response. The linear regression model is :

$$y_i = \sum_{j=1}^{p} \beta_j x_{ij} + \eta_i, \tag{B.1}$$

where η_i is a noise following some probabilistic distribution.

Assume that the predictors are standardized and that the response is centered. The Lasso estimate is then given by :

$$\hat{\boldsymbol{\beta}}^L(\lambda) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}} \left[\sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \|\boldsymbol{\beta}\|_1 \right], \tag{B.2}$$

with λ a non-negative scalar that determines the level of the constraints which is user-provided. We remark that :

— When $\lambda = 0$, $\hat{\boldsymbol{\beta}}^L$ is an ordinary least square estimation.

— When $\lambda = +\infty$, we get $\hat{\boldsymbol{\beta}}^L = \mathbf{0}_p$.

The Lasso estimate for linear regression has two main advantages :

1. it allows dealing with ill-posed problems where the number of observations is inferior to the number of variables,
2. it allows performing variable selection : for a proper choice of λ, $\hat{\boldsymbol{\beta}}^L(\lambda)$ will be parsimonious.

The Lasso estimate for linear regression can also be written in the following form :

$$\hat{\boldsymbol{\beta}}^L(\lambda) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p \quad \|\boldsymbol{\beta}\|_1 \leqslant \tilde{\lambda}}{\operatorname{argmin}} \left[\sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right]. \tag{B.3}$$

These two formulations (equation (B.2) which is the penalized formulation and equation (B.3) which is the constrained formulation) are equivalent in the sense that for each non negative λ there is a non-negative $\tilde{\lambda}$ leading to the same solution.

Model for network reverse-engineering

Suppose that we have selected N genes across T time points and for P individuals ; we note x_{npt} the expression of gene n for individual p at time-point t. Since each gene will be considered exactly once as a response variable, our model is composed of N linear regression models. As the action of a gene of another is not instantaneous, we define :

$$\tilde{\boldsymbol{x}}_{np.} = \begin{pmatrix} x_{npt_2} \\ \vdots \\ x_{npt_T} \end{pmatrix} \quad \text{and} \quad \check{\boldsymbol{x}}_{np.} = \begin{pmatrix} x_{npt_1} \\ \vdots \\ x_{npt_{T-1}} \end{pmatrix},$$

$$\tilde{\boldsymbol{x}}_{n..} = \begin{pmatrix} \tilde{\boldsymbol{x}}_{n1.} \\ \vdots \\ \tilde{\boldsymbol{x}}_{nP.} \end{pmatrix} \quad \text{and} \quad \check{\boldsymbol{x}}_{n..} = \begin{pmatrix} \check{\boldsymbol{x}}_{n1.} \\ \vdots \\ \check{\boldsymbol{x}}_{nP.} \end{pmatrix}.$$

We note that $\tilde{\boldsymbol{x}}_{np.}$ begins at time point t_2 and ends at time point t_T, while $\check{\boldsymbol{x}}_{np.}$ begins at time point t_1 and ends at time point t_{T-1}. In the following, when gene n is the response variable we will use $\tilde{\boldsymbol{x}}_{np.}$, and $\check{\boldsymbol{x}}_{np.}$ when gene n is a predictor variable.

We further assume that each gene has been assigned to one and only one of the T time-cluster (one cluster for each time).

We have previously proposed (Vallat *et al.* (2013)) the follwing linear regression model :

$$\tilde{\boldsymbol{x}}_{n..} = \sum_{n'=1}^{N} \boldsymbol{F}_{m(n')m(n)} \omega_{n'n} \check{\boldsymbol{x}}_{n'..} + \boldsymbol{\varepsilon}_n,$$

where :

— $m(\bullet)$ is the function that maps a gene to its time-cluster,
— $\boldsymbol{F}_{m(n')m(n)}$ is a $T-1$ square matrix that describes the action of genes,
— $\omega_{n'n}$ is the strength of the connection from gene i toward gene j,
— $\boldsymbol{\varepsilon}$ is a noise vector of length $T-1$ with $\mathbb{E}(\boldsymbol{\varepsilon}) = 0$ and $\text{var}(\boldsymbol{\varepsilon}) = \sigma^2$

We choose to use a Lasso estimate for our linear regression model :

$$(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{F}}) = \underset{\substack{\omega_{n'n} \in \mathbb{R},\ 1 \leqslant n', n \leqslant N \\ \boldsymbol{F}_{ab} \in \mathcal{M}_{T-1}(\mathbb{R}), 1 \leqslant a, b \leqslant T}}{\text{argmin}} \left[\sum_{n=1}^{N} \left(\tilde{\boldsymbol{x}}_{n..} - \sum_{n'=1}^{N} \boldsymbol{F}_{m(n')m(n)} \omega_{n'n} \check{\boldsymbol{x}}_{n'..} \right)^2 \right],$$

with the constraint :

$$\forall n = 1, ..., N, \quad \sum_{n'=1}^{N} \omega_{n'n} \leqslant \lambda_n.$$

So, $\tilde{\boldsymbol{x}}_{n..}$ is the regulated gene and $\boldsymbol{x}_{n'..}, n' = 1, \cdots, N$ are the regulators. Notice that matrix $F_{m(n')m(n)}$ permits to the link between genes n' and n to evolve across time. To enforce temporal causality we need the two following time constraints :

1. $m(n') \geqslant m(n) \Rightarrow \boldsymbol{F}_{m(n')m(n)} = 0$: this ensures that a gene with temporal cluster t_k can influence a gene with temporal cluster $t_{k'}$ if and only if $k < k'$,
2. the matrices $\boldsymbol{F}$ are lower triangular matrices : this ensures that the expression of a gene at time t_k can influence another gene at time $t_{k'}$ if and only if $k < k'$.

Sub-diagonals and the diagonal of matrices F are supposed to be invariant (Vallat *et al.* (2013)). Consequently, interactions depend only on time index differences rather than absolute time index.

We solve this problem with a coordinate ascent approach, by iteratively supposing the $\boldsymbol{F}$ matrices or the $\omega_{n'n}$ matrices known. The result of the optimization is a connectivity network described by the nonzero elements of $\hat{\omega}_{n'n}(obs)$ combined with a set of cluster-dependent interaction models described by the set $\hat{\boldsymbol{F}}_{m(n')m(n)}(obs)$.

However, if clusters are sufficiently homogeneous, inference of matrices $\boldsymbol{F}_{m(n')m(n)}$ doesn't depend on which genes are active (*i.e.* which $\omega_{n'n} \neq 0$). That's why a non iterative algorithm is proposed in which estimation of of matrices $F_{m(i)m(j)}$ precedes estimation of matrix $\boldsymbol{\Omega}$.

To get a more robust result, at each step, the estimation of matrices $\boldsymbol{F}_{m(n')m(n)}$ is done several times throughout cross-validation. Furthermore, to avoid computational issues, the new solution is chosen by a linear combination between the old and the new solution.

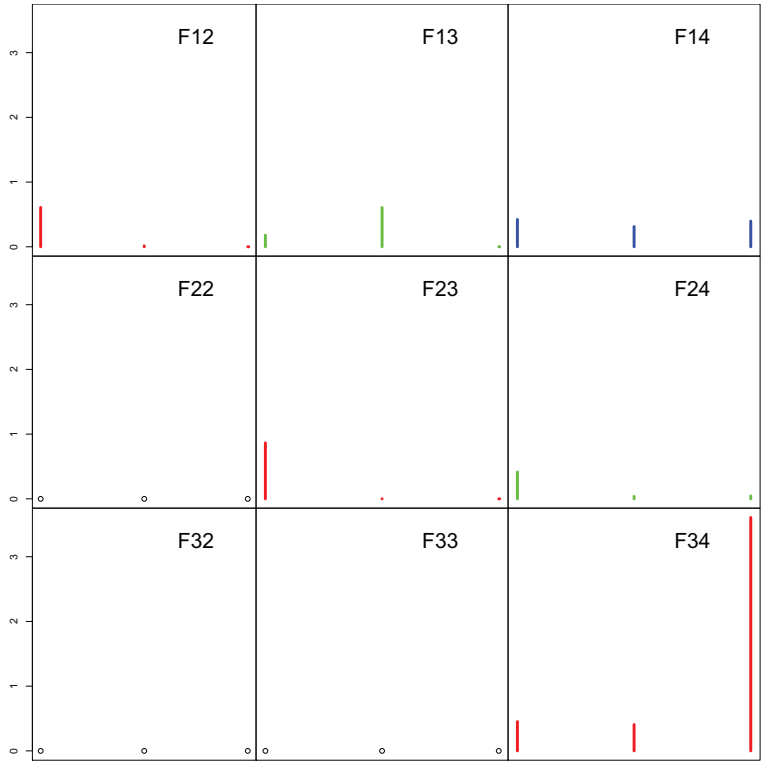
B.5.2 Performing the reverse-engineering algorithm

To perform this algorithm on our data :

```
> network<-inference(Selection)
```

```
We are at step :  1
The convergence of the network is (L1 norm) : 0.01096
We are at step :  2
The convergence of the network is (L1 norm) : 0.00302
We are at step :  3
The convergence of the network is (L1 norm) : 0.00217
We are at step :  4
The convergence of the network is (L1 norm) : 0.00177
We are at step :  5
The convergence of the network is (L1 norm) : 0.00146
We are at step :  6
The convergence of the network is (L1 norm) : 0.00111
We are at step :  7
The convergence of the network is (L1 norm) : 0.00089
```

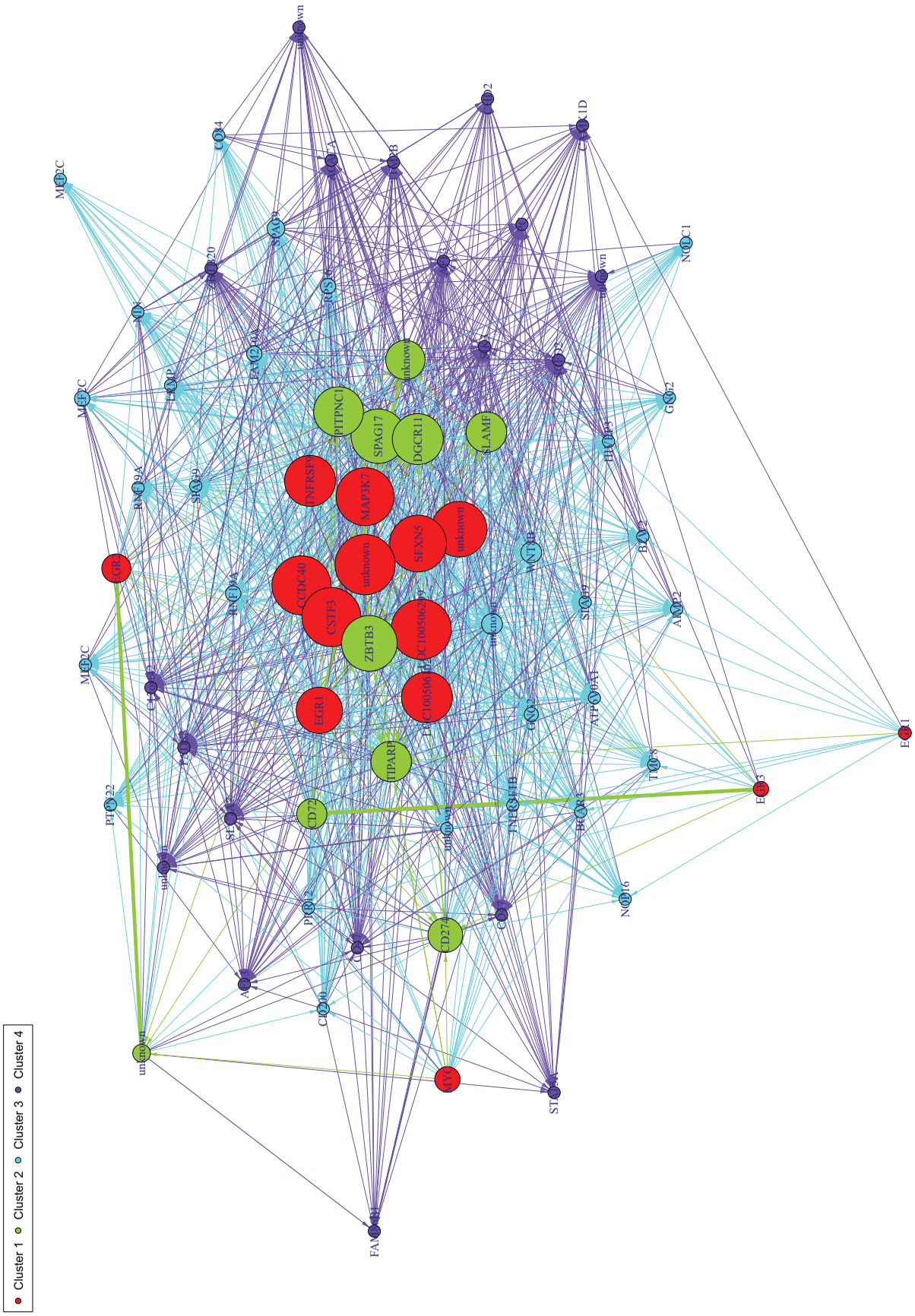
We can plot a representation of F matrices (Figure B.5) and the resulting network (Figure B.6) by simply using the `plot` method :



Supplemental Figure B.5 – *The F matrices; for each matrix, the first bar plot corresponds to the coefficient of the diagonal, the second to the first sub-diagonal...*

```
> plot(network,choice="F")
> plot(network,choice="network",gr=Selection@group,label_v=Selection@name)
```

Note that all network plots are computed using the Igraph R package (Csardi et Nepusz (2006)).



Supplemental Figure B.6 – *The resulting network with all edges*

The number of edges in the network makes the message difficult to interpret ; and as we will see in the next section, results in term of predictive positive value and F-score can be improved when choosing a right cutoff level. Using the `nv` option, we will choose a cutoff under which the regression coefficient estimates ($\hat{\omega_{ij}}(obs)$) are set to 0. In Figure B.7 a cutoff of 0.11 is chosen.

B.5.3 Choosing the best cutoff for edge minimal strength

The difficulty is now to choose the best cutoff. As a starting point, we propose method `evolution`, that allows the user to see, in a html page, the evolution of the network when the cutoff is growing up. When the `fix` option is set to `FALSE`, at each step the position of the genes are re-calculated.

```
> evolution(network,seq(0,0.4,by=0.01),gr=Selection@group,
   fix=TRUE,label_v=Selection@name)
> evolution(network,seq(0,0.4,by=0.01),gr=Selection@group,
   fix=FALSE,label_v=Selection@name)
```

To see the result of these functions, go to :

— `http://www-irma.u-strasbg.fr/~njung/evolution_fix_true/evol.html` : here the `fix` option is set to `TRUE`.

— `http://www-irma.u-strasbg.fr/~njung/evolution_fix_false/evol.html` : here the `fix` option is set to `FALSE`.

As it is mostly accepted, gene regulatory networks are supposed to be scale-free (Jeong *et al.* (2000)). The notion of scale freeness in networks relies on the probability distribution of the number of outgoing edges. A network is called scale free when this distribution is a power law distribution (Clauset *et al.* (2009)). As this family of law is large, it is difficult to test such an hypothesis. We used the test proposed by Clauset *et al.*(Clauset *et al.* (2009)) :

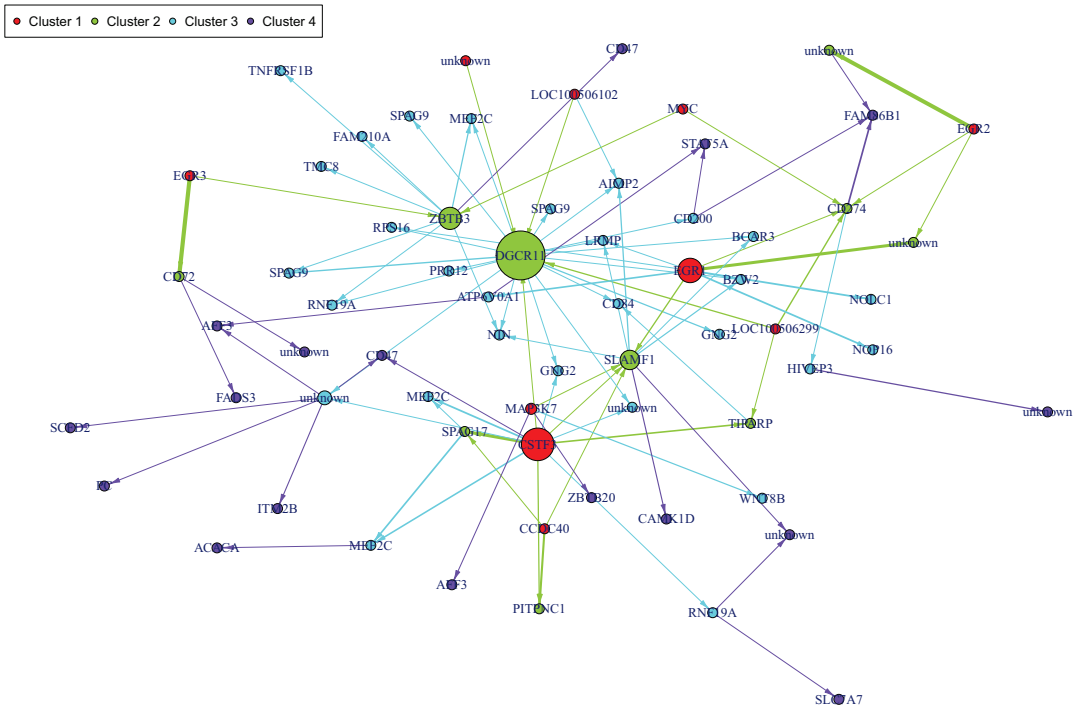
```
> #To be computed:
> #evol_cutoff<-cutoff(network)
> nv<-0.15
```

We plot here the smooth interpolation rather than the exact values, as our interest relies mostly on the trend (Figure B.8). We propose a choice of cutoff that relies on two criteria :

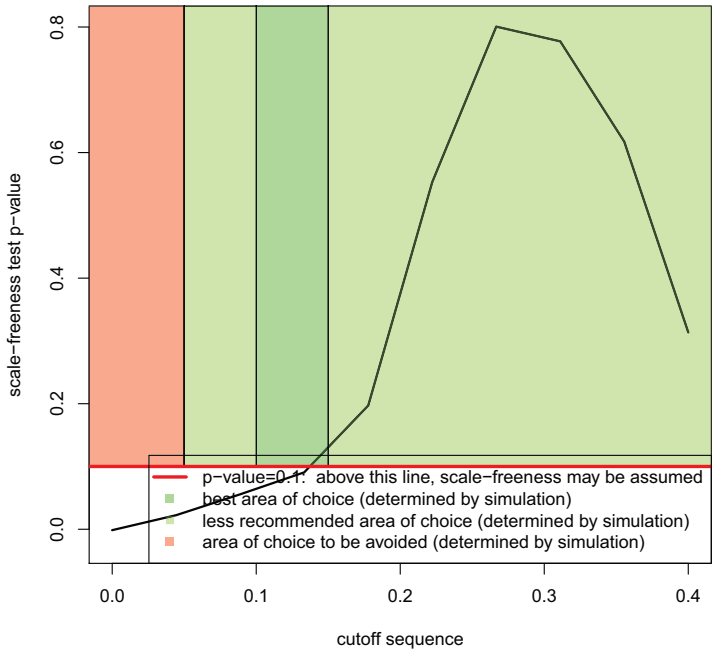
— the p-value should be greater than 0.10 : in this case, the scale-freeness of the network is reliable (Clauset *et al.* (2009)).

— we determined by simulation the best area of choice (on the plot (Figure B.8)).

Based on these two criteria, we choose a cutoff of $nv = 0.11$.



Supplemental Figure B.7 – *The resulting network with a cutoff of 0.15*



Supplemental Figure B.8 – *Evolution of scale-freeness of the network in function of the cutoff. The p-value corresponds to the adequacy of the data to a power law distribution.*

B.5.4 Analyzing the network

One may want to know which genes are important in the network. In our representation, the bigger the vertex the larger the number of outgoing edges. Indeed, genes with many outgoing edges, the hubs, are important in the network. But genes controlling these hubs should be considered with attention. The `analyze_network` method allows computing different indicators :

— betweenness : it is a measure of the node centrality. It is calculated, for node n, by the following formula :

$$\sum_{s \neq t \neq n} \frac{\sigma_{st}(n)}{\sigma_{st}}$$

where σ_{st} is the number of shortest ways between s and t, and $\sigma_{st}(n)$ is the number of shortest ways between s and t passing by n ;

— degree : the number of outgoing edges ;

— output : the sum of weights of outgoing genes ;

— closeness : it is a measure of the distance (in terms of shortest path) of a gene to others.

As our network is weighted we used specific measures developed by Opsahl (Opsahl *et al.* (2010)).

```
> analyze<-analyze_network(network,nv,Selection@name)
> head(analyze)
          node betweenness degree    output closeness
1 LOC100506299           0      3 0.8133348 14.841838
2       CCDC40           0      3 0.8884602  7.305208
3      unknown           0      1 0.1749376  8.826222
4 LOC100506102           0      2 0.3661906  9.622533
5      TNFRSF9           0      0 0.0000000  0.000000
6        CSTF3           0     12 3.4345058 23.065564
```

Note that one can plot the network and modulate the size of the vertex following one of this measure, using the `weight.node` option.

Using again the package `animation`, we can see how the signal spreads in the network by turning to `TRUE` the option `ani` :

```
> plot(network,nv=nv,gr=Selection@group,ani=TRUE,label_v=Selection@name,
   edge.arrow.size=0.9,edge.thickness=1.5)
```

Result is available at `http://www-irma.u-strasbg.fr/~njung/network_spread/spread.html`.

The method `plot` has basically two steps :

1. it calculates the position of the vertex,
2. it plots the graph.

In some case, it is interesting to produce two plots of a same network without changing vertex positions. Here is a way to do that, using the `ini` option of method plot :

```
> P<-position(network,nv=nv)
> #plotting the network with the given position
> plot(network,nv=nv,gr=Selection@group,ini=P,label_v=Selection@name)
```

However, we didn't develop all possibilities of the `plot` option ; for more possibilities, please refer to the manual :

```
> vignette("Cascade-manual")
```

B.6 Prediction of gene expression modulations after a knock-out experiment

Once the network has been reverse-engineered, we want to know the impact of an experimental perturbation in this network. For example, what would happen if expression of EGR1 is knocked-out ?

```
> EGR1<-which(Selection@name %in% "EGR1")
```

First the `geneNeighborhood` method allows determining which are the neighborhood of EGR1 (see Figure B.9).

```
> geneNeighborhood(network,targets=EGR1,nv=nv,ini=P,
   label_v=Selection@name)
> #label.hub: only hubs vertex should have a name
> #label_v: name of the vertex
```

We predict gene expression modulations within the network if EGR1 is experimentaly knocked-out.

```
> prediction_ko5<-predict(Selection,network,nv=nv,targets= EGR1)
```

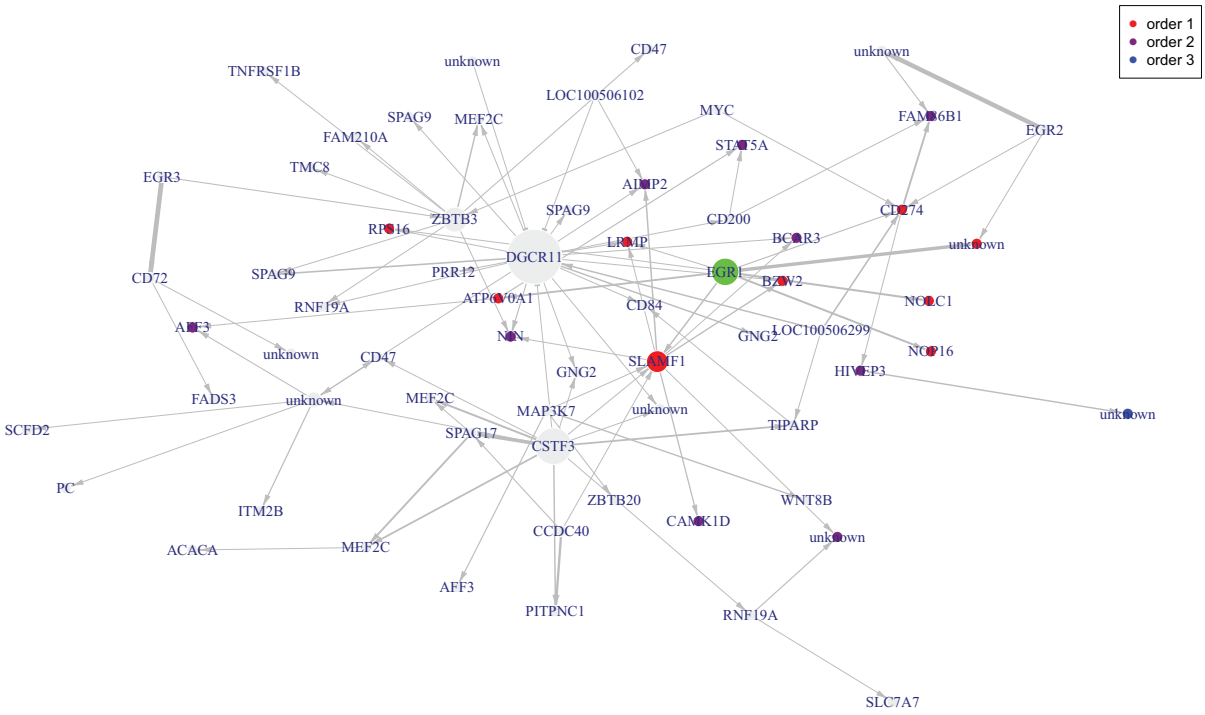
Then we plot the results (Figure B.10) :

```
> #We plot the results.
> #Here for example we see changes at time point t2:
> plot(prediction_ko5,time=2,ini=P,label_v=Selection@name)
```

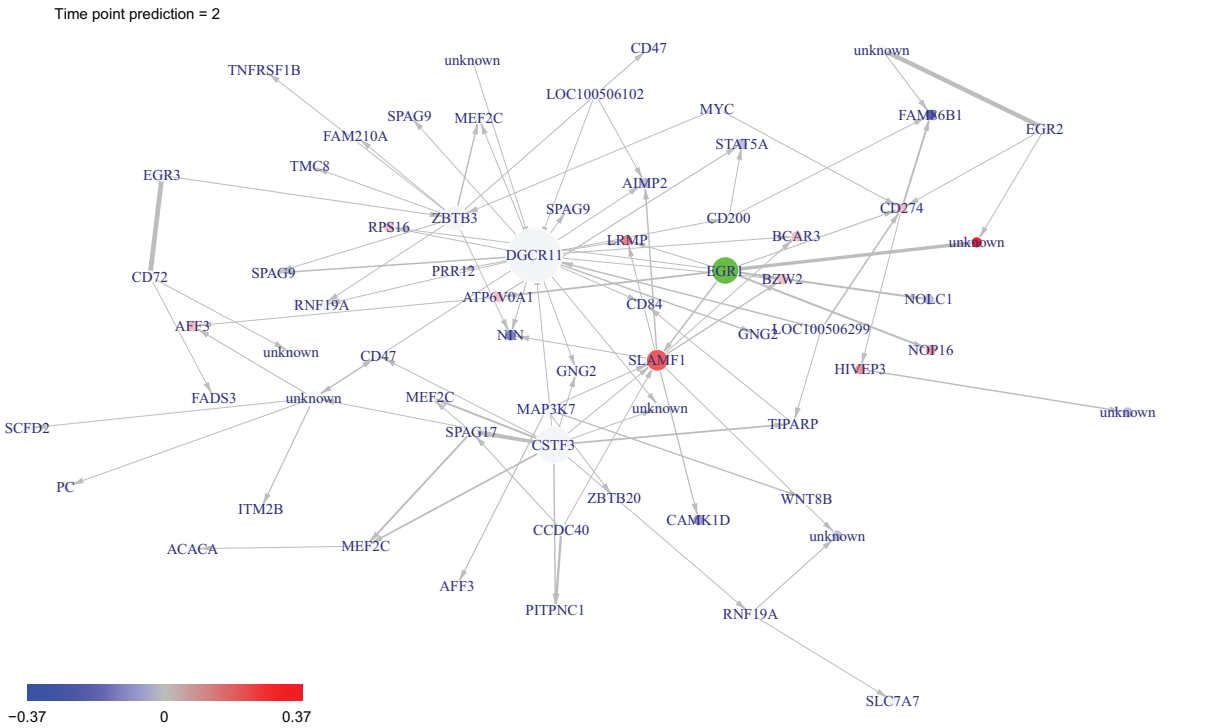
B.7 Simulation

To simulate gene expressions based on a gene regulatory network, we first have to generate the network. Here, we implemented an algorithm that is inspired by the *preferential attachment* from Barabási (Barabási (2002), Jeong *et al.* (2003)). We adapted this algorithm in our case of temporal cascade networks.

We then use our linear model to make some simulations, using Laplace laws to initiate the algorithm.



Supplemental Figure B.9 – *Neighborhood of gene EGR1*

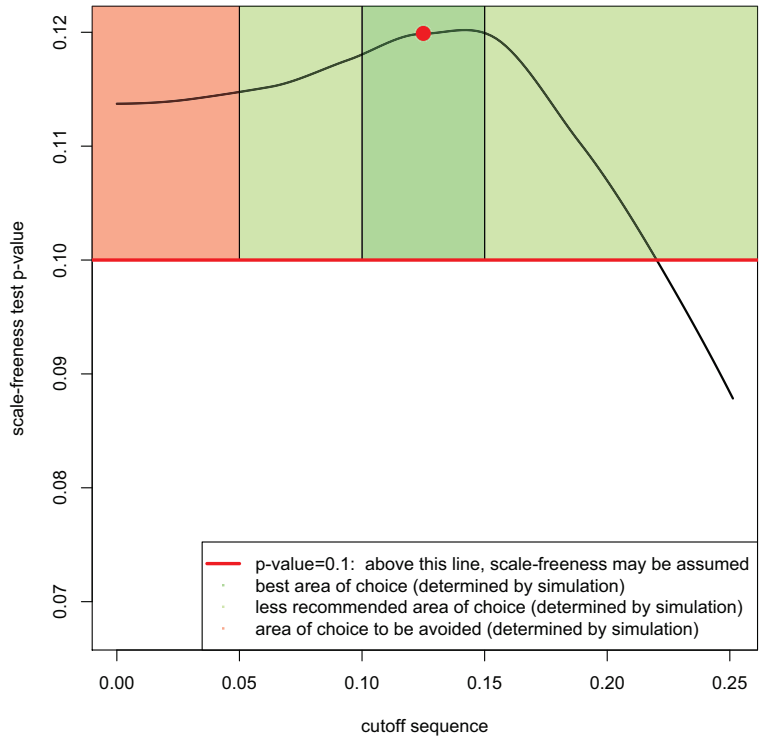


Supplemental Figure B.10 – *Perturbation modulation at time point 2 of the network consecutively to the knock-out of EGR1.*

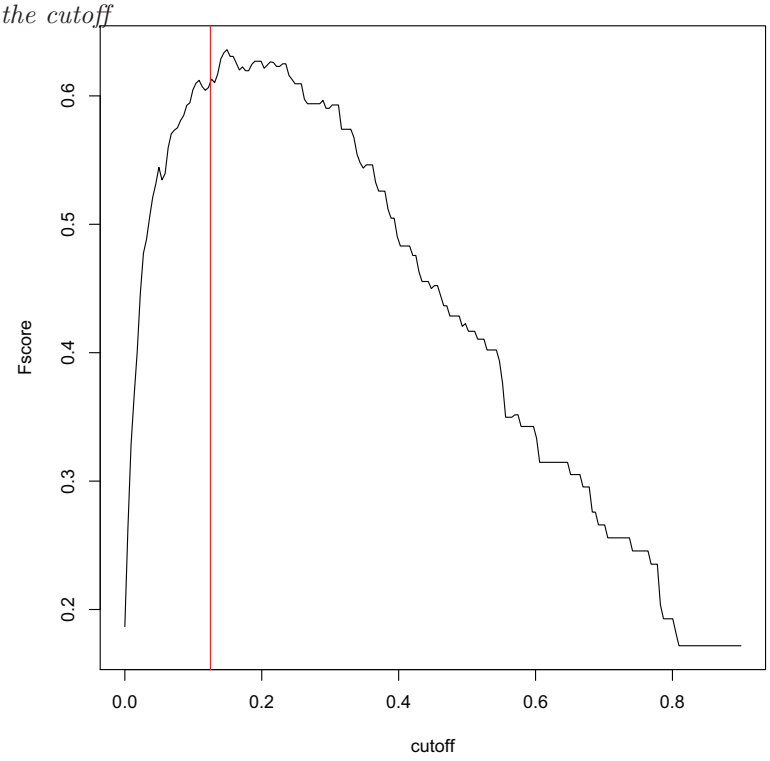
```
> #We set the seed to make the results reproducible
> set.seed(1)
> #We create a random scale free network
> Net<-network_random(
          nb=100,
          time_label=rep(1:4,each=25),
          exp=1,
          init=1,
          regul=round(rexp(100,1))+1,
          min_expr=0.1,
          max_expr=2,
          casc.level=0.4
          )
> #We change F matrices
> T<-4
> F<-array(0,c(T-1,T-1,T*(T-1)/2))
> for(i in 1:(T*(T-1)/2)){diag(F[,,i])<-1}
> F[,,2]<-F[,,2]*0.2
> F[2,1,2]<-1
> F[3,2,2]<-1
> F[,,4]<-F[,,2]*0.3
> F[3,1,4]<-1
> F[,,5]<-F[,,2]
> Net@F<-F
> #We simulate gene expression according to the network Net
> M<-gene_expr_simulation(
          network=Net,
          time_label=rep(1:4,each=25),
          subject=5,
          level_pic=200)

> #We infer the new network
> Net_inf<-inference(M)

> #Comparing true and inferred networks
> F_score<-rep(0,200)
> #Here are the cutoff level tested
> test.seq<-seq(0,max(abs(Net_inf@network*0.9)),length.out=200)
> u<-0
> for(i in test.seq){
          u<-u+1
          F_score[u]<-compare(Net,Net_inf,i)[3]
   }

> #Choosing the cutoff
> cut.seq<-cutoff(Net_inf)
> points(0.125,0.1199,col="red",pch=16,cex=2)
> #Corresponding Fscore evolution
> plot(test.seq,F_score,type="l",xlab="cutoff",ylab="Fscore")
> abline(v=0.125,col="red")
```

Figure B.11 shows the evolution of the p-value of the scale-freeness test while Figure B.12 shows the corresponding evolution of the F-score. As shown, choosing the best cut-off allows a dramatic increase of the cut-off.

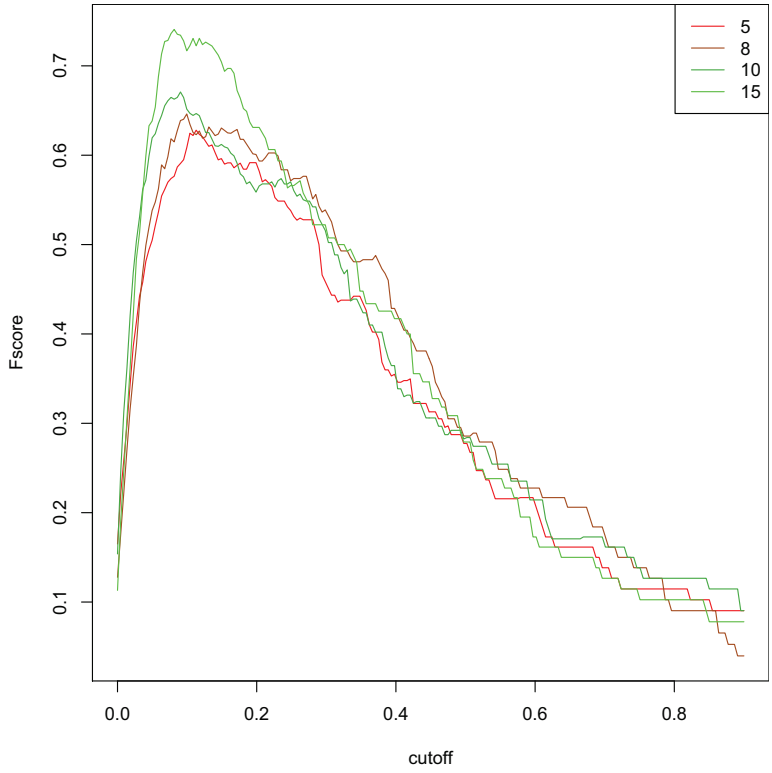
Figure B.13 show the evolution of the F-score when the number of individuals increase.



Supplemental Figure B.11 – *Evolution of the scale-freeness of the network in function of the cutoff*



Supplemental Figure B.12 – *Evolution of F-score in function of the cutoff*



Supplemental Figure B.13 – *Evolution of F-score in function of the cutoff and the number of subject in the study*

Application 2

C

C.1 Introduction

We perform a reanalysis of "E-MTAB-1475" dataset (den Ham *et al.* (2013)). This dataset is composed of temporal gene expressions from stimulated lymphocytes. The aim of the paper is to uncover the specific transcription that sustains the specific cellular T-lymphocyte differentiation in Th1 or Th2 T-lymphocyte cells. The experimental design is a follow :

- Gene expression of T-lymphocyte naïve cells (without any stimulation). Not time-repeated (this measure is replicated four times).
- Gene expression of T-lymphocyte cells after a neutral stimulation. These measurements are done after one, two, three and four days (each measure is replicated three times).
- Gene expression of T-lymphocyte cells differentiating themselves in Th1 T-lymphocyte cells following a neutral stimulation + a Th1 oriented stimulation. These measurements are done after one, two, three and four days (each measure is replicated three times).
- Gene expression of T-lymphocyte cells differentiating themselves in Th2 T-lymphocyte cells following a neutral stimulation + a Th2 oriented stimulation. These measurements are done after one, two, three and four days (each measure is replicated three times).

First, we configured our gene selection function with the same parameters as in the original paper (den Ham *et al.* (2013)) : false discovery rate (FDR) set to 0.05, minimal log-fold change (LFC) to 1.5 and we discarded the use of patterns in our package. As expected, we found the differentially expressed genes already highlighted in this paper.

Next, we focus on the cellular differentiation transcriptional program of the Th1 T-lymphocyte cells by studying the differentially expressed genes between the Neutral stimulation + Th1 versus the Neutral stimulation. We then reanalyzed the dataset taking advantage from the gene selection function of our package, according to their temporal patterns and differential expressions.

C.2 Importing the "E-MTAB-1475" dataset

The microarray datasets are stored online under the reference "E-MTAB-1475" at :

http://www.ebi.ac.uk/arrayexpress/

Before running the following code lines, you first have to download the appropriate dataset in your current working folder.

```
> library(ArrayExpress)
> MTAB1475 = getAE("E-MTAB-1475", type = "full",local=TRUE)
> cnames = getcolproc(MTAB1475);
> MTAB1475proc = procset(MTAB1475, cnames[2])
> #The columns order should follow the Cascade format order:
>
>
> exprsNeu <- exprs(MTAB1475proc)[,c(8,17,25,33,9,2,26,34,10,18,27,35)]
> exprsTh1 <- exprs(MTAB1475proc)[,c(11,19,28,36,12,20,3,4,13,21,29,37)]
> exprsTh2 <- exprs(MTAB1475proc)[,rep(c(14,15,16),rep(4,3))+c(0,8,16,24)]
> exprsNaive <- exprs(MTAB1475proc)[,c(1,5,6,7)]
```

Data are then converted in the Cascade format :

```
> library(Cascade)
> Neu_ma<-as.micro_array(exprsNeu,c(1,2,3,4),3)
> Th1_ma<-as.micro_array(exprsTh1,c(1,2,3,4),3)
> Th2_ma<-as.micro_array(exprsTh2,c(1,2,3,4),3)
> Naive<-as.micro_array(exprsNaive,c(1),4)
> M<-list(Naive,Neu_ma,Th1_ma,Th2_ma)
```

M is a list with all "micro_array" data corresponding to the four different conditions : naïve (control without any stimulation), neutral stimulation, neutral+TH1 oriented stimulation, neutral+TH2 oriented stimulation.

C.3 Gene selections and reverse-engineering of the networks

C.3.1 Note on the "geneSelection" function

The geneSelection function may be used specifying either a stimulated `micro_array` object and an unstimulated `micro_array` object (the control), either a list of `micro_array` objects and a list specifying the contrasts. In the second case, contrasts are specified using a list with the following specific form :

First element : "condition", "condition&time" or "pattern". The "condition" specification is used when the overall goal is to compare two conditions. The "condition&time" specification is used when comparing two conditions at two precise time points. The "pattern" specification allows to choose at which time points selected genes should be expressed or not.

Second element : a vector of length 2, corresponding to the two conditions that should be compared. If a non-temporal dataset is used as control, it should be the first element of the `micro_array` list and the option "`cont`=TRUE" should be used.

Third element : depends on the first element. This element is not needed if "condition" has been specified. If "condition&time" has been specified, then this is a vector containing the time point at which the comparison should be done. If "pattern" has been specified, then this is a vector of 0 and 1 of length T, where T is the number of time points. Time points where differential expression is wanted are provided with 1.

C.3.2 Selecting differentially expressed genes without specifying patterns

In the first part of our analysis, we configured our method with the same parameters as in the paper of van den Ham et al. (den Ham *et al.* (2013)) : FDR set to 0.05, minimal log-fold change to 1.5 and we discarded the use of patterns.

We define the following contrasts :

- neutral versus naive
- Th1 versus naive
- Th2 versus naive
- Th1 versus neutral
- Th2 versus neutral
- Th1 versus Th2

```
> naive_neutre<- list("condition",c(1,2))
> naive_th1<- list("condition",c(1,3))
> naive_th2<- list("condition",c(1,4))
> neutre_th1<- list("condition",c(2,3))
> neutre_th2<- list("condition",c(2,4))
> th2_th1<- list("condition",c(4,3))
```

We now use these same contrasts, but we focus on each day separately. Note : as a convention, when the control is not time-measured, it is considered that measures are done at day 0 (e.g. naïve condition).

```
> naive_neutre_d1<- list("condition&time",c(1,2),c(0,1))
> naive_neutre_d2<- list("condition&time",c(1,2),c(0,2))
> naive_neutre_d3<- list("condition&time",c(1,2),c(0,3))
> naive_neutre_d4<- list("condition&time",c(1,2),c(0,4))
> naive_th1_d1<- list("condition&time",c(1,3),c(0,1))
> naive_th1_d2<- list("condition&time",c(1,3),c(0,2))
> naive_th1_d3<- list("condition&time",c(1,3),c(0,3))
> naive_th1_d4<- list("condition&time",c(1,3),c(0,4))
> naive_th2_d1<- list("condition&time",c(1,4),c(0,1))
> naive_th2_d2<- list("condition&time",c(1,4),c(0,2))
> naive_th2_d3<- list("condition&time",c(1,4),c(0,3))
> naive_th2_d4<- list("condition&time",c(1,4),c(0,4))
> neutre_th1_d1<- list("condition&time",c(2,3),c(1,1))
> neutre_th1_d2<- list("condition&time",c(2,3),c(2,2))
> neutre_th1_d3<- list("condition&time",c(2,3),c(3,3))
> neutre_th1_d4<- list("condition&time",c(2,3),c(4,4))
> neutre_th2_d1<- list("condition&time",c(2,4),c(1,1))
```

```
> neutre_th2_d2<- list("condition&time",c(2,4),c(2,2))
> neutre_th2_d3<- list("condition&time",c(2,4),c(3,3))
> neutre_th2_d4<- list("condition&time",c(2,4),c(4,4))
> th2_th1_d1<- list("condition&time",c(4,3),c(1,1))
> th2_th1_d2<- list("condition&time",c(4,3),c(2,2))
> th2_th1_d3<- list("condition&time",c(4,3),c(3,3))
> th2_th1_d4<- list("condition&time",c(4,3),c(4,4))
```

We proceed to the corresponding gene selections, setting the option "cont" to TRUE (because the control is not time-measured) and lfc (minimal log fold change) to 1.5 :

```
> S_naive_neutre<-geneSelection(M,naive_neutre,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_th1<-geneSelection(M,naive_th1,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_th2<-geneSelection(M,naive_th2,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_neutre_th1<-geneSelection(M,neutre_th1,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_neutre_th2<-geneSelection(M,neutre_th2,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_neutre_d1<-geneSelection(M,naive_neutre_d1,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_neutre_d2<-geneSelection(M,naive_neutre_d2,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_neutre_d3<-geneSelection(M,naive_neutre_d3,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_neutre_d4<-geneSelection(M,naive_neutre_d4,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_th1_d1<-geneSelection(M,naive_th1_d1,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_th1_d2<-geneSelection(M,naive_th1_d2,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_th1_d3<-geneSelection(M,naive_th1_d3,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_th1_d4<-geneSelection(M,naive_th1_d4,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_th2_d1<-geneSelection(M,naive_th2_d1,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_th2_d2<-geneSelection(M,naive_th2_d2,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_th2_d3<-geneSelection(M,naive_th2_d3,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_naive_th2_d4<-geneSelection(M,naive_th2_d4,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_neutre_th1_d1<-geneSelection(M,neutre_th1_d1,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_neutre_th1_d2<-geneSelection(M,neutre_th1_d2,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_neutre_th1_d3<-geneSelection(M,neutre_th1_d3,-1,cont=TRUE,alpha=0.05,
```

```
   data_log=FALSE,lfc=1.5)
> S_neutre_th1_d4<-geneSelection(M,neutre_th1_d4,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_neutre_th2_d1<-geneSelection(M,neutre_th2_d1,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_neutre_th2_d2<-geneSelection(M,neutre_th2_d2,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_neutre_th2_d3<-geneSelection(M,neutre_th2_d3,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_neutre_th2_d4<-geneSelection(M,neutre_th2_d4,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_th2_th1_d1<-geneSelection(M,th2_th1_d1,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_th2_th1_d2<-geneSelection(M,th2_th1_d2,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_th2_th1_d3<-geneSelection(M,th2_th1_d3,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> S_th2_th1_d4<-geneSelection(M,th2_th1_d4,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
```

We can use the `org.Mm.eg.db` Bioconductor database which is provided as a package to find gene names :

```
> library(org.Mm.eg.db)
> #Here S is the union of all selections
>
> S<-unionMicro(list(S_naive_neutre,S_naive_th1,
   S_naive_th2,S_neutre_th1,S_neutre_th2,
   S_naive_neutre_d1,S_naive_neutre_d2,S_naive_neutre_d3,S_naive_neutre_d4,
   S_naive_th1_d1,S_naive_th1_d2,S_naive_th1_d3,S_naive_th1_d4,
   S_naive_th2_d1,S_naive_th2_d2,S_naive_th2_d3,S_naive_th2_d4,
   S_neutre_th1_d1,S_neutre_th1_d2,S_neutre_th1_d3,S_neutre_th1_d4,
   S_neutre_th2_d1,S_neutre_th2_d2,S_neutre_th2_d3,S_neutre_th2_d4,
   S_th2_th1_d1,S_th2_th1_d2,S_th2_th1_d3,S_th2_th1_d4
   ))
> ff<-function(x){substr(x, 1, nchar(x)-3)}
> ff<-Vectorize(ff)
> #Here is the function to transform the probeset names to gene ID.
>
>
>
> probe_to_id<-function(n){
   x <- (org.Mm.egENSEMBL2EG)
   mp<-mappedkeys(x)
   xx <- unlist(as.list(x[mp]))
   genes_all = xx[ff(n)]
   indi2<-!is.na(genes_all)
   genes_all=genes_all[!is.na(genes_all)]
   gns <- unlist(mget((unlist(genes_all)), org.Mm.egSYMBOL) )
   Name<-rep("na",length(n))
   Name[ indi2]<-gns
```

```
   return(toupper(Name))
   }
> Name<-probe_to_id(S@name)
```

As expected, we found the differentially expressed genes already highlighted in this paper (den Ham *et al.* (2013)). Here is the list :

- Batf
- Batf2
- Batf3
- Gata3
- Ifng
- Il13
- Il4
- Tbx21 (Tbet)
- Ccl17
- Ccl24
- Egr2
- Foxp3
- Hopx
- Il10
- Il6
- Il9
- Mecom
- Mycn
- Nfil3
- Rbpj
- Spic
- Stad1
- Stat5a
- Tnf
- Tnfrsf4
- Vdr.

The whole list of genes contains 1.454 elements (see supplemental file genelists.xls available at `http://www-math.u-strasbg.fr/genpred/sites/genpred/IMG/xls/genelists.xls`).

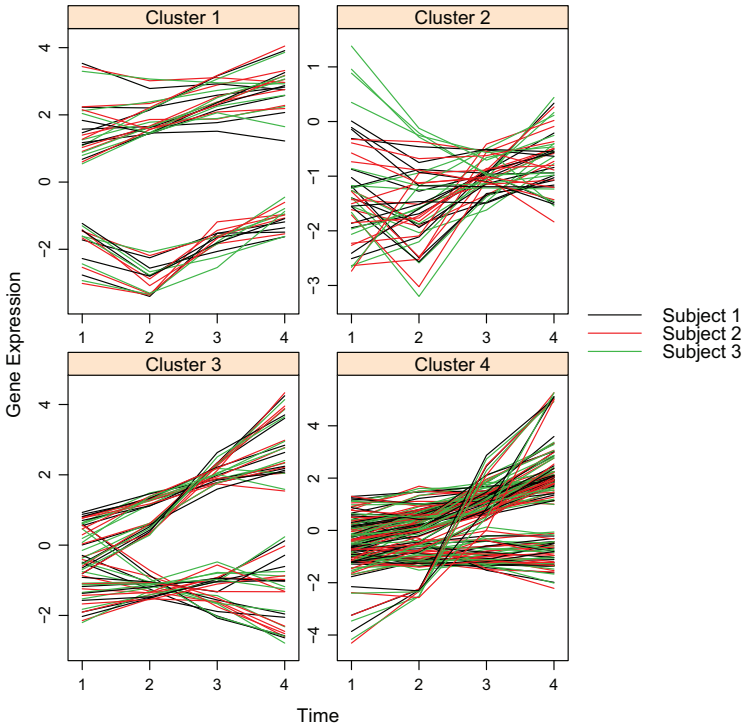
C.3.3 Selecting differentially expressed genes with specific patterns

To reverse-engineer the TH1 specific network, we select genes that are differentially expressed with the neutral+TH1 stimulation versus the neutral stimulation. Here we use the same methodology than in the paper of Vallat *et al.* (Vallat *et al.* (2013)), that's why we select the following patterns :

- early differentially expressed genes (D1,D2,D1+D2),
- late differentially expressed genes (D3,D4,D3+D4),
- gene with high differential expression (D1+D2+D3+D4).

```
> neutre_th1_early<- list("patterns",c(2,3),
   rbind(c(1,1,0,0),c(1,0,0,0),c(0,1,0,0)))
> neutre_th1_late<- list("patterns",c(2,3),
   rbind(c(0,0,1,1),c(0,0,1,0),c(0,0,0,1)))
> neutre_th1_high<- list("patterns",c(2,3),
   rbind(c(1,1,1,0),c(1,1,1,1)))
> th1_early<-geneSelection(M,neutre_th1_early,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> th1_late<-geneSelection(M,neutre_th1_late,-1,cont=TRUE,alpha=0.05,
   data_log=FALSE,lfc=1.5)
> th1_high<-geneSelection(M,neutre_th1_high,-1,cont=TRUE,
   alpha=0.05,data_log=FALSE,lfc=1.5)
> dim(th1_early)
[1] 15 12
> dim(th1_late)
[1] 66 12
> dim(th1_high)
[1] 13 12
```

```
> selection_th1<-unionMicro(list(th1_early,th1_late,th1_high))
> S<-selection_th1
> plot(selection_th1)
> S_name<-probe_to_id(S@name)
> selection_th1@name<-S_name
```



Supplemental Figure C.1 – *Selected genes for inference of the TH1 specific network*

Interestingly, many genes of interest that are specific to TH1 are in our short selection of 94 genes :

- IFNG
- GATA3
- IL13
- IL4
- FOXP3
- HOPX
- IL9
- MECOM
- MYCN

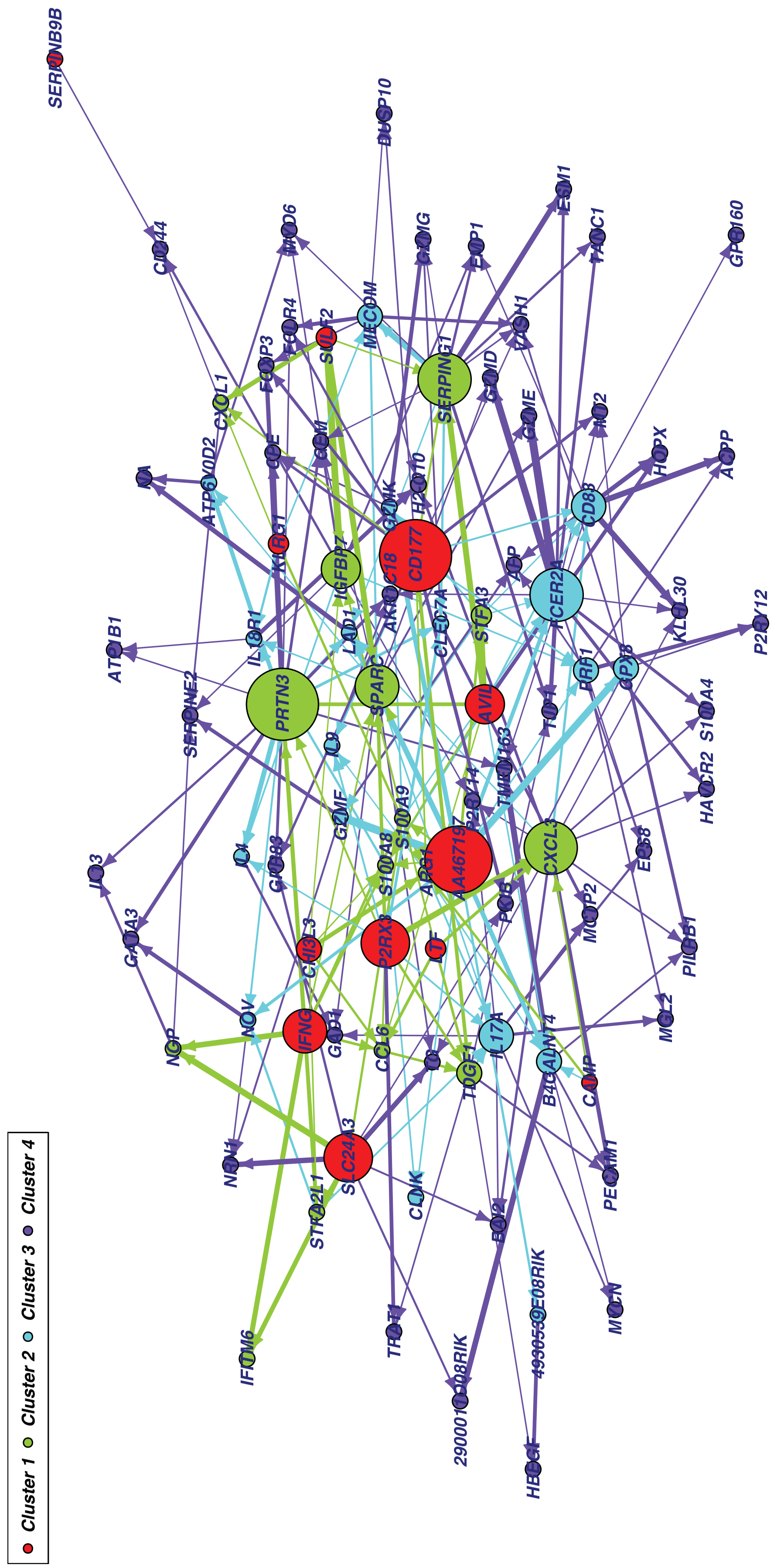
The Figure C.1 shows the selected genes grouped within time-clusters.

C.3.4 Reverse-engineering the TH1 specific network

Using our selection of 94 genes, we reverse-engineer the network.

```
> network_th1<-inference(selection_th1)
> cutoff(network_th1)
> #To save the plot of the network in a convenient way,
> #we use the Cairo package (Simon Urbanek and Jeffrey Horner, 2012)
>
> library(Cairo)
> Cairo(40, 20, file="th1.pdf",type="pdf",units="cm",bg="white")
```

```
> par(mar=c(0,0,0,0))
> pos<-position(network_th1,nv=0.12)
> plot(network_th1,nv=0.12,gr=selection_th1@group,label_v=S_name,ini=pos,
   edge.arrow.size=0.85,edge.thickness=2)
> dev.off()
```



Supplemental Figure C.2 – *Result of the reverse-engineering of the TH1 specific network.*

The TH1 specific network is represented in Figure C.2.

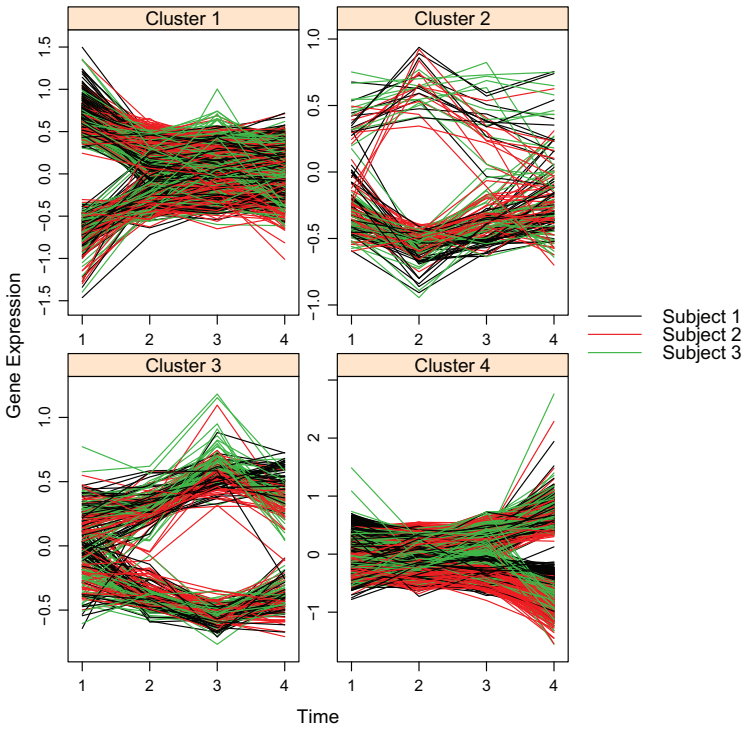
We note that the key TH1 gene, IFNG, is a hub in our network. Furthermore, the analyze of the network shows that IFNG is the third most important gene in our network by its closeness score.

```
> D.influence<-analyze_network(network_th1,nv=0.12,network_th1@name)
> head(D.influence[rev(order(D.influence$closeness)),])
```

C.3.5 Using very specific patterns and lower the log-fold change threshold

Up to now, we decrease the minimal log fold change parameter from 1.5 to 0.5. To enhance the gene selection, we decide to look for very specific patterns. Such a procedure is known to reduce the false-positive rate (Di Camillo *et al.* (2012)).

```
> contrastTH1d1<-  list("patterns",c(1,3),rbind(c(1,0,0,0)))
> contrastTH1d2<-  list("patterns",c(1,3),rbind(c(0,1,0,0)))
> contrastTH1d3<-  list("patterns",c(1,3),rbind(c(0,0,1,0)))
> contrastTH1d4<-  list("patterns",c(1,3),rbind(c(0,0,0,1)))
> S1<-geneSelection(M,contrastTH1d1,-1,cont=TRUE,
   alpha=0.05,data_log=FALSE,lfc=0.5)
> S2<-geneSelection(M,contrastTH1d2,-1,cont=TRUE,
   alpha=0.05,data_log=FALSE,lfc=0.5)
> S3<-geneSelection(M,contrastTH1d3,-1,cont=TRUE,
   alpha=0.05,data_log=FALSE,lfc=0.5)
> S4<-geneSelection(M,contrastTH1d4,-1,cont=TRUE,
   alpha=0.05,data_log=FALSE,lfc=0.5)
> S<-unionMicro(list(S1,S2,S3,S4))
```

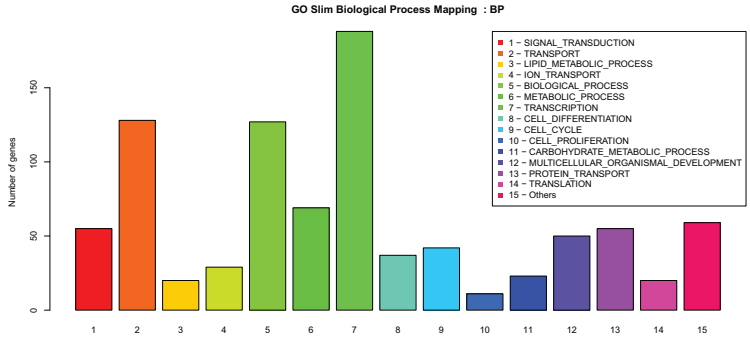


Supplemental Figure C.3 – *New gene selection with a minimal log fold-change set to 0.5 and very specific temporal patterns.*

As shown in Figure C.3, this new selection presents very specific time patterns in which each gene has a peak of differential expression at only one time point.

Additionally, we can analyze the predominant biological gene functions (GO) of the selected genes using two more libraries (geneListPie and org.Mm.eg.db) :

```
> library(geneListPie)
> library(org.Mm.eg.db)
> data(goslim.mouse.BP)
> genes_all<-probe_to_id(S@name)
> pdf("functions_TH1.pdf",width=14)
> r1<-geneListProfile(goslim.mouse.BP, genes_all, threshold=10)
> labels<-sub("_", "__", r1$labels)
> labels<-sub(".*__", "", labels)
> barplot(r1$sizes, main="GO Slim Biological Process Mapping  : BP",
   col=rainbow(length(r1$sizes)),names.arg=1:length(r1$sizes),
   ylab="Number of genes")
> legend("topright",legend=paste(paste(1:15,"-"),labels),
   pch=15,col=rainbow(length(r1$sizes)))
> dev.off()
```



Supplemental Figure C.4 – *Biological functions of the selected genes.*

We notice that this allows selecting supplemental genes harboring "significant" patterns, enriching the gene selection with transcriptional factors and key genes in the lymphocyte biology (IL15, IL6, IL7R, JAK2, IKZF3 or ITGAL for example) (see Figure C.4).

C.3.6 Reverse-engineering an additional network

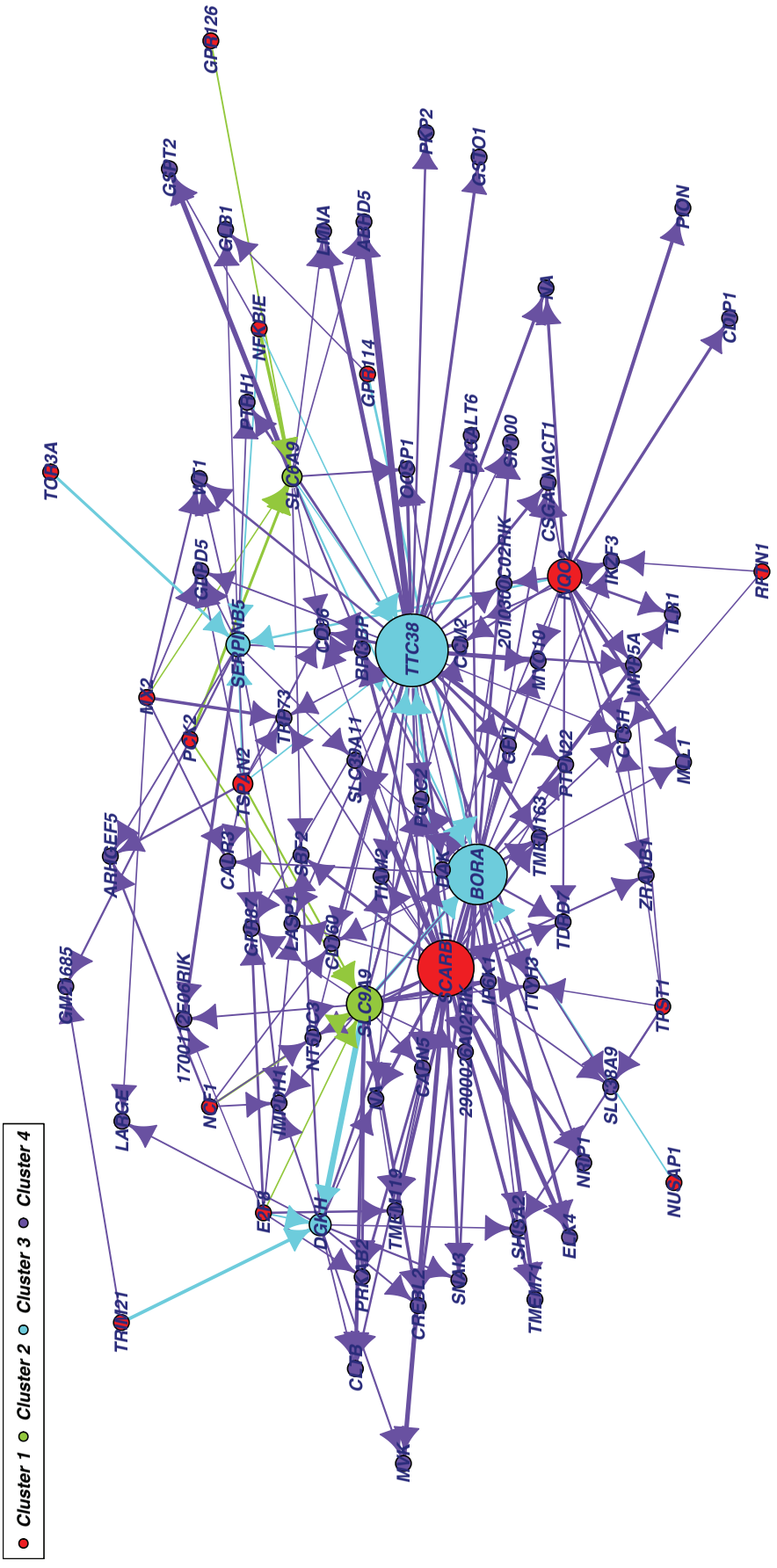
For readability, we only use 5% of the selected genes from each previous sub-selection (S1, S2, S3, S4) :

```
> S1_res<-geneSelection(M,contrastTH1d1,0.05,cont=TRUE,
   alpha=0.05,data_log=FALSE,lfc=0.5)
> S2_res<-geneSelection(M,contrastTH1d2,0.05,cont=TRUE,
   alpha=0.05,data_log=FALSE,lfc=0.5)
> S3_res<-geneSelection(M,contrastTH1d3,0.05,cont=TRUE,
   alpha=0.05,data_log=FALSE,lfc=0.5)
> S4_res<-geneSelection(M,contrastTH1d4,0.05,cont=TRUE,
   alpha=0.05,data_log=FALSE,lfc=0.5)
> S_res<-unionMicro(list(S1_res,S2_res,S3_res,S4_res))
```

```
> network_resTH1<-inference(S_res)
> cutoff(network_resTH1)
```

The resulting network is presented in Figure C.5.

```
> Cairo(40, 20, file="th1_res.pdf",type="pdf",units="cm",bg="white")
> par(mar=c(0,0,0,0))
> pos2<-position(network_resTH1,nv=0.12)
> plot(network_resTH1,nv=0.12,gr=S_res@group,label_v=probe_to_id(S_res@name),
   edge.arrow.size=0.73,edge.thickness=2)
> dev.off()
```

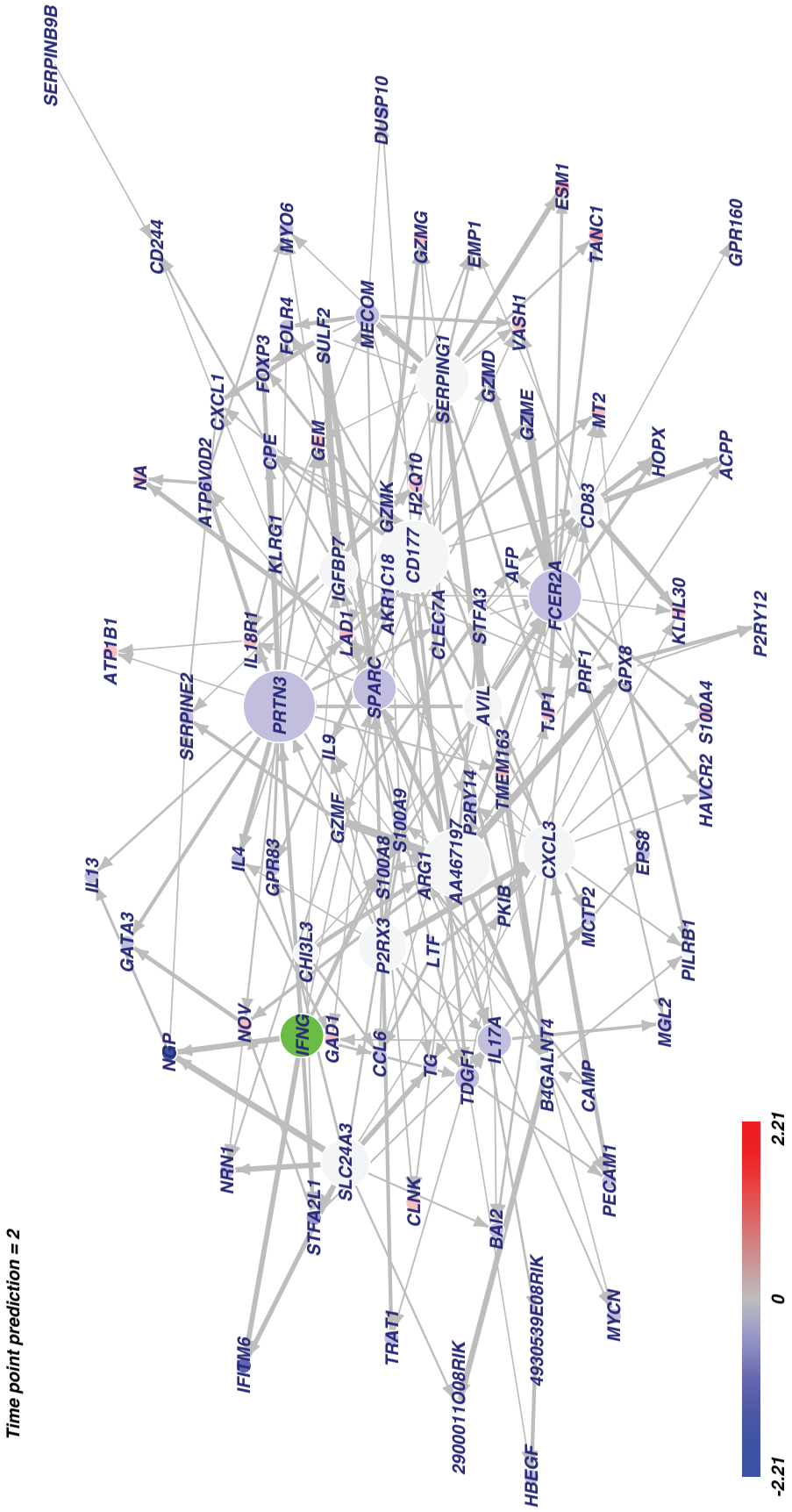


Supplemental Figure C.5 – *Result of the reverse-engineering of the second network.*

C.4 Prediction of gene expressions after a knock-out experiment

The package further allows predicting the result of a knock-out experiment on the downstream genes in the network. Here we choose to knock-out IFN-G in our first TH1 specific network. The Figure 6 shows the effects of the knock-out experiment on IFN-G.

```
> IFNG<-which(selection_th1@name %in% "IFNG")
> network.p<-predict(selection_th1,network_th1,nv=0.12,targets=IFNG)


> Cairo(40, 20, file="th1_pred.pdf",type="pdf",units="cm",bg="white")
> par(mar=c(0,0,0,0))
> plot(network.p,time=2,label_v=S_name,ini=pos,
   edge.arrow.size=0.85,edge.thickness=2)
> dev.off()
```



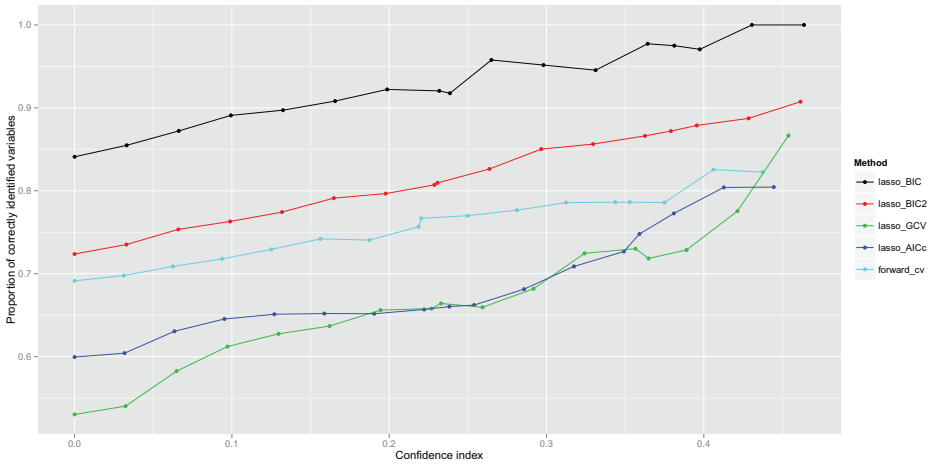
Supplemental Figure C.6 – *Predicted effects of the knock-out of IFN-G.*

Supplement information for Chapter 3

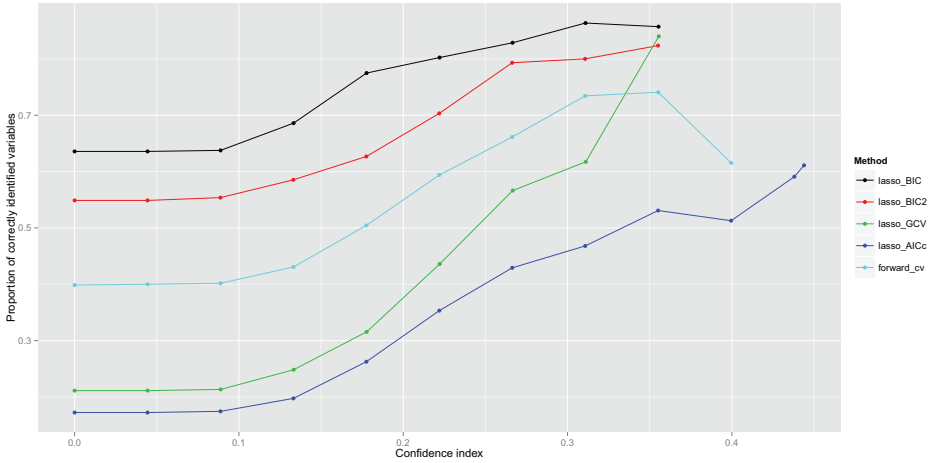
D

D.1 Confidence index

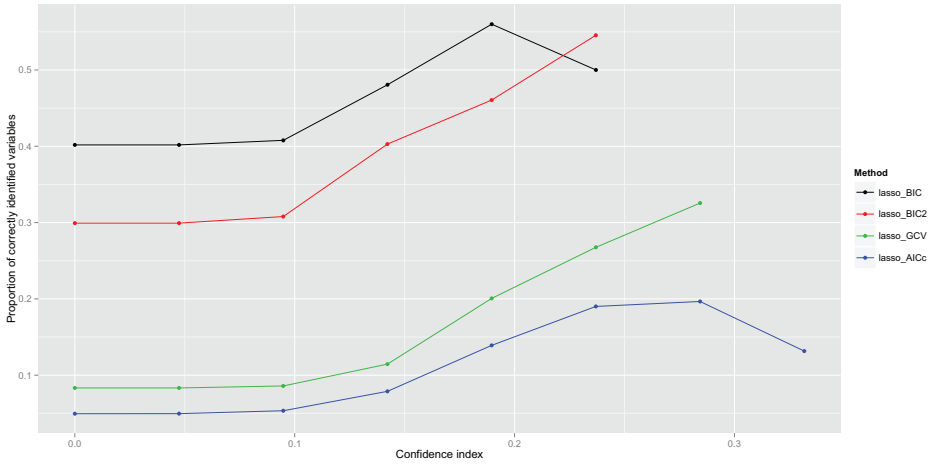
For the four simulation situations, we plot the proportion of correctly identified variables against the confidence index.



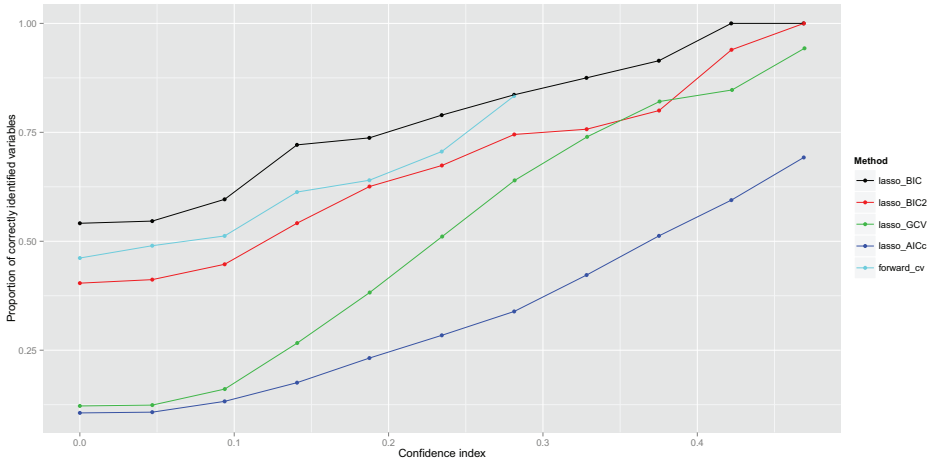
Supplemental Figure D.1 – *Situation 1*



Supplemental Figure D.2 – *Situation 2*



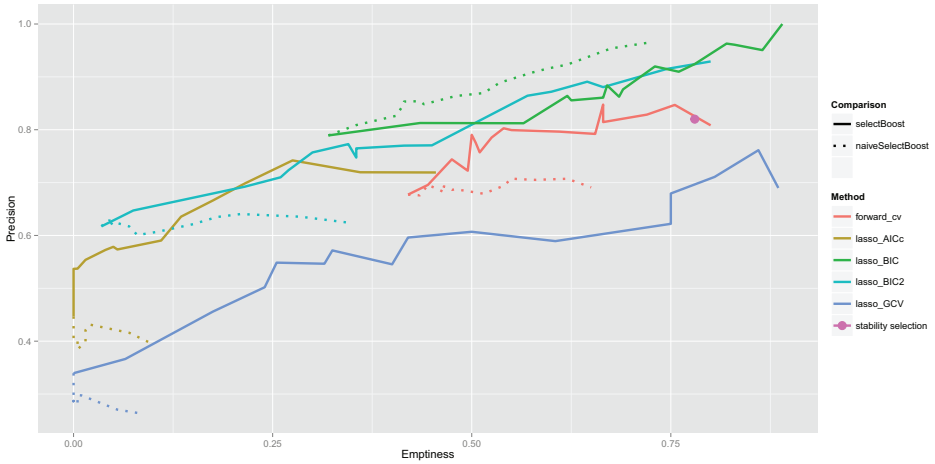
Supplemental Figure D.3 – *Situation 3*



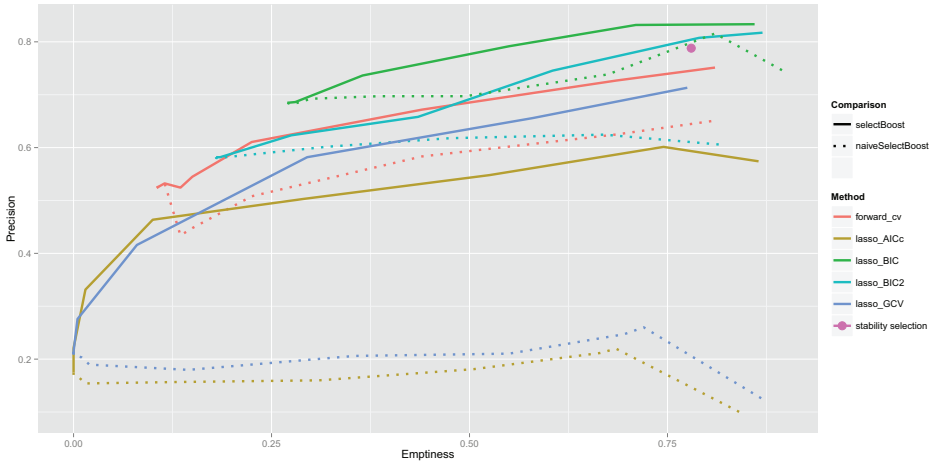
Supplemental Figure D.4 – *Situation 4*

D.2 Comparisons

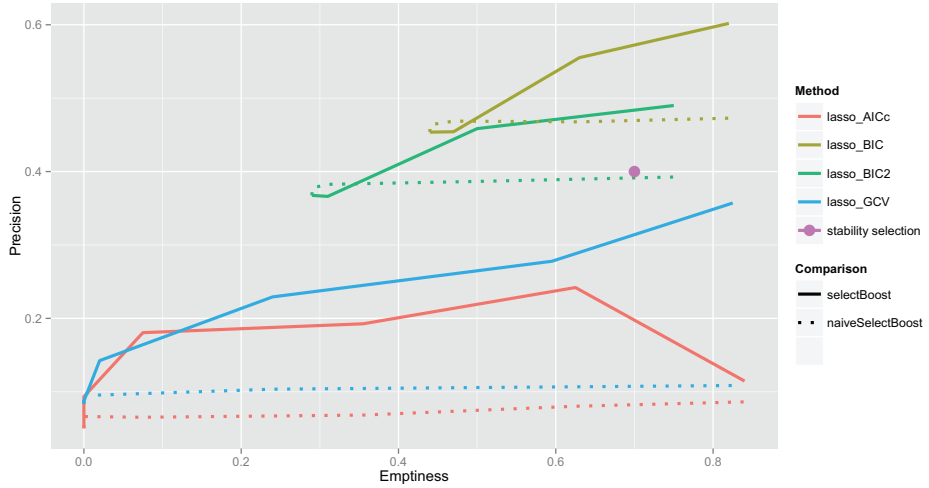
For the four simulation situations, we show the performance of our algorithm against the performance of the naive selectBoost algorithm and the Stability Selection algorithm. The best algorithm is the one with the lowest proportion of empty models with the highest precision.



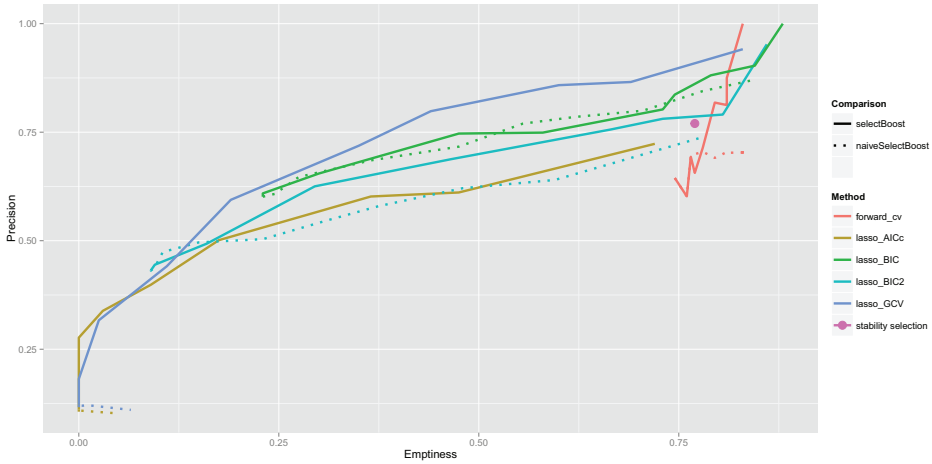
Supplemental Figure D.5 – *Situation 1*



Supplemental Figure D.6 – *Situation 2*



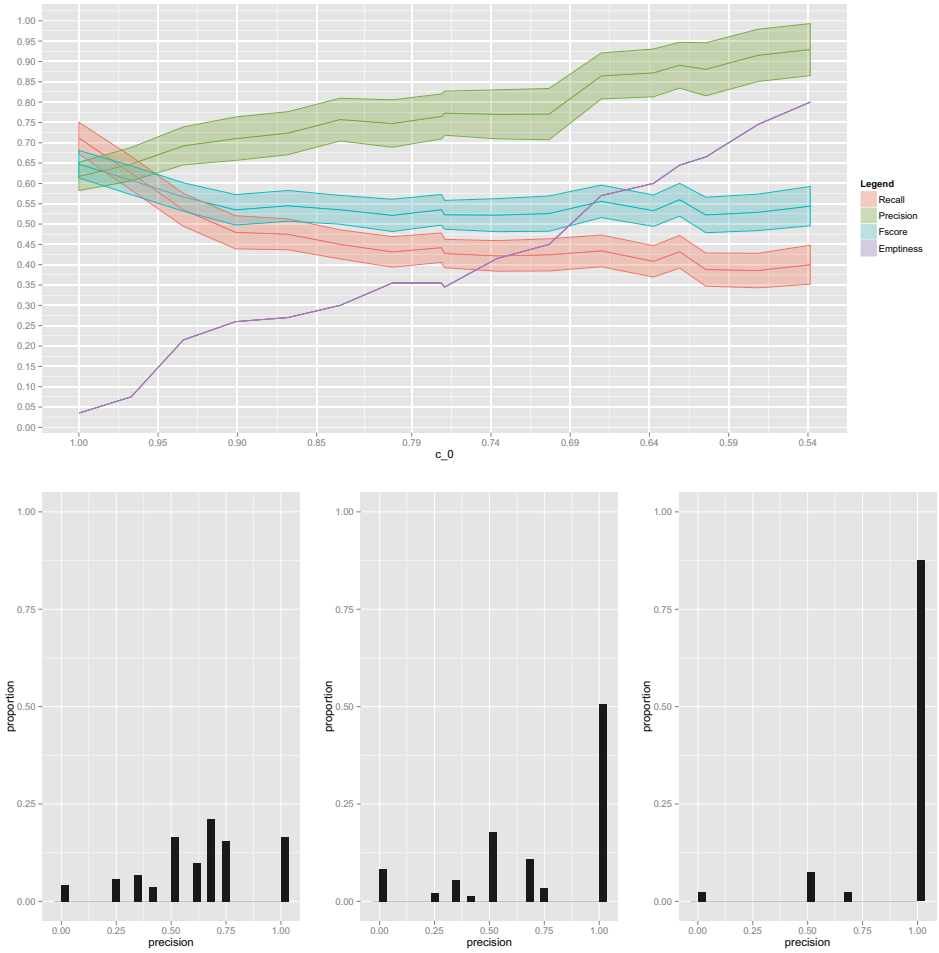
Supplemental Figure D.7 – *Situation 3*



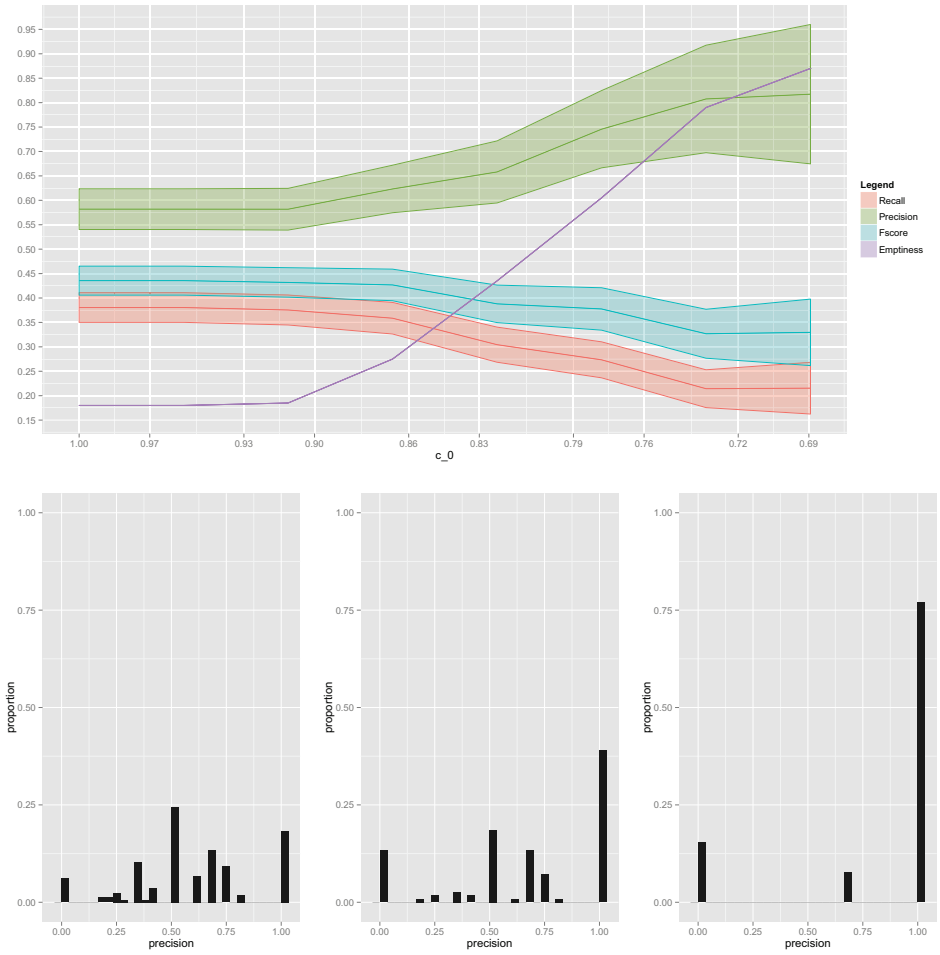
Supplemental Figure D.8 – *Situation 4*

D.3 Example of results : modified BIC (BIC2)

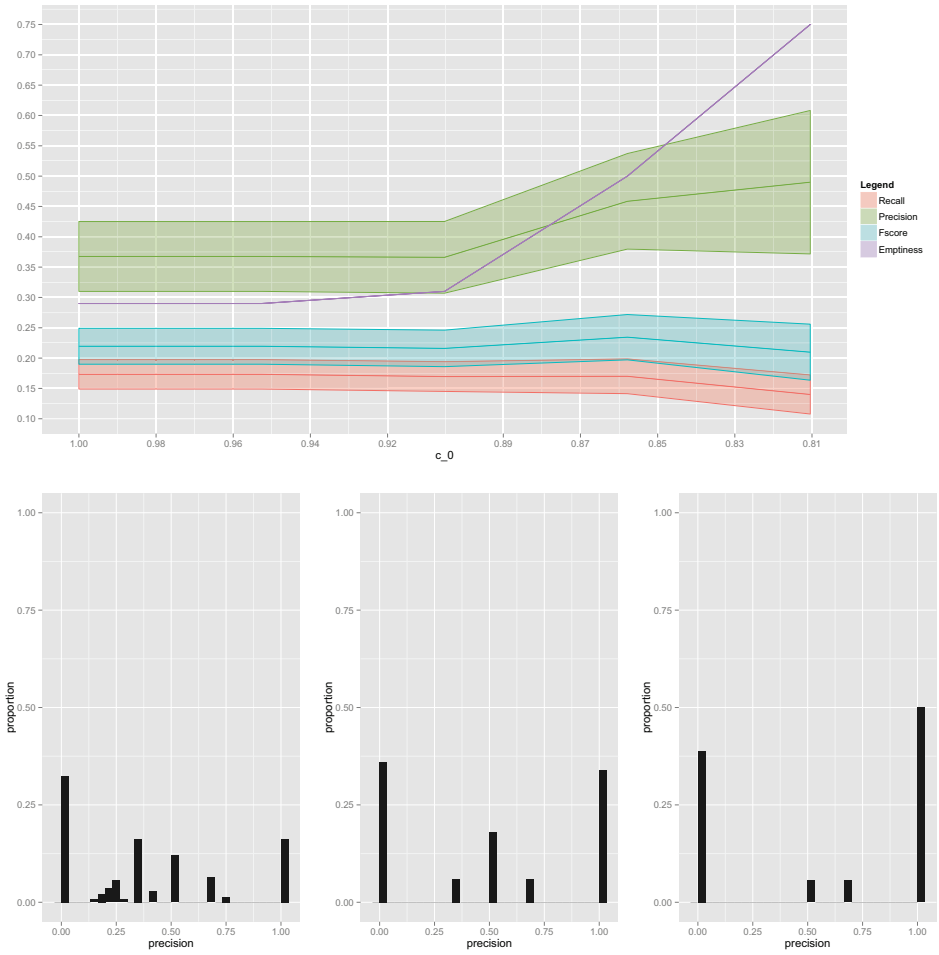
In this section we show the evolution of recall, precision, Fscore and proportion of empty models with 95% confidence interval in function of the c_0 parameter. We also show three histograms with the evolution of the distribution of the precision for three c_0 (see legends)



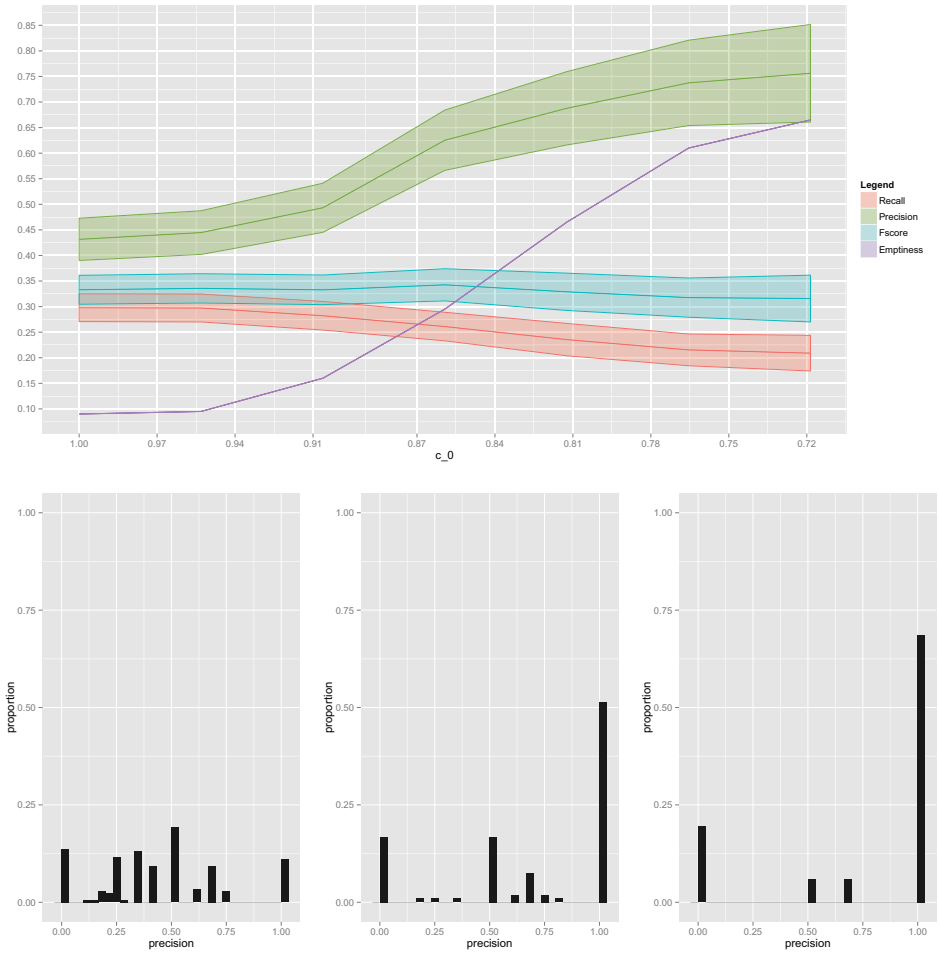
Supplemental Figure D.9 – *Situation 1. The histograms show the evolution of the distribution of the precision. From left to right :* $c_0 = 1, 0.79, 0.54$



Supplemental Figure D.10 – *Situation 2. The histograms show the evolution of the distribution of the precision. From left to right :* $c_0 = 1, 0.79, 0.69$



Supplemental Figure D.11 – *Situation 3. The histograms show the evolution of the distribution of the precision. From left to right :* $c_0 = 1, 0.9, 0.81$



Supplemental Figure D.12 – *Situation 4. The histograms show the evolution of the distribution of the precision. From left to right :* $c_0 = 1, 0.87, 0.72$

%includeannexe4

Réseaux multi-états

E

Cette annexe présente le résumé qui a été soumis pour les journées SFdS dans lequel nous développons l'idée de réseaux multi-états.

Inférence conjointe de réseaux de gènes dans de multiples états

Nicolas Jung [1,2], Myriam Maumy-Bertrand [1], Laurent Vallat [2] & Frédéric Bertrand [1]

[1] *Institut de Recherche en Mathématique Avancée (IRMA), Strasbourg* [2] *Institut d'Hématologie, Faculté de Médecine de Strasbourg*

Résumé. Quand une cellule est stimulée, le programme génique qu'elle contient est activé. Les gènes mis en action apportent alors une réponse concertée au stimulus. Cette réponse est modélisée statistiquement par un réseau dans lequel les noeuds correspondent aux gènes et les liens correspondent à leurs interactions. À partir des expressions de ces gènes, un nombre important de méthodes statistiques a été proposé pour inférer les réseaux de gènes sous-jacents.

Certaines maladies, comme le cancer par exemple, affectent le programme génique. Pour tenter de comprendre les perturbations qui en résultent, il est nécessaire d'estimer le réseau de gènes dans les différents états (sain/malade, par exemple). En supposant que seule une partie restreinte du réseau est affectée par la maladie, une estimation simultanée des différents réseaux (correspondant chacun à un état particulier) est nécessaire.

Cette estimation simultanée permet d'une part d'utiliser pleinement l'information commune dans les sous-parties du réseau inchangées d'un état à l'autre, et d'autre part, d'obtenir des réseaux plus facilement comparables. La méthode que nous proposons s'inscrit dans ce cadre.

Mots-clés. Réseau de régulation de gènes, Sélection de variables, Méthodes de classification.

Abstract. When a signal triggers a cell, the inherent genetic program is activated, leading to a concerted action of stimulated genes. This response is modeled statistically thanks to a network in which nodes correspond to genes and links correspond to potential interactions. Based on these gene expressions, lots of methods have been proposed to reverse-engineer underlying gene networks.

Some diseases, such as in cancer for example, modify the genetic program. In order to understand which perturbations are linked to these modifications, it is necessary to reverse-engineer the gene network in the different states (eg., healthy/ill). Assuming that the part of the network which is affected by the disease is restricted, a simultaneous reverse-engineering procedure might be necessary.

This would allow to use the common information contained in the part of the network that is not affected by the disease. Futherthemore, this estimation leads to comparable networks.

Keywords. Gene regulatory networks, Variable selection, Clustering.

1 Introduction et motivations

Quand une cellule est stimulée, le programme génique qu'elle contient est activé. Les gènes mis en action apportent alors une réponse concertée au stimulus. Cette réponse est modélisée statistiquement par un réseau dans lequel les noeuds correspondent aux gènes et les liens correspondent à leurs interactions. Depuis l'introduction de technologies à haut débit qui permettent de mesurer simultanément l'expression de milliers de gènes, beaucoup de méthodes statistiques ont été proposées pour l'inférence de ces réseaux de régulation.

Ces méthodes peuvent être regroupées en trois catégories principales. Il y a d'abord les méthodes dites d'interactions, dans lesquelles une mesure de proximité entre les gènes est définie, comme l'entropie dans la méthode ARACNe de Margolin et al. (2006). Parmi ces méthodes, se trouve la classe des GGMs (Graphical Gaussian Models), dans laquelle l'hypothèse de normalité permet de calculer le cœfficient de corrélation partiel linéaire (Chiquet (2011), par exemple). Ces méthodes sont relativement peu coûteuses en temps de calcul, mais elles ne permettent pas de décrire la dynamique des systèmes biologiques. Nous trouvons ensuite les méthodes dites d'optimisation dans lesquelles il convient de distinguer les réseaux booléens d'une part (Liang et al. (1998)), et les réseaux bayésiens d'autre part (Dondelinger et al. (2011)). Dans ces derniers, l'ensemble des gènes régulateurs d'un gène donné est appelé *parents*. Des probabilités *a priori* de chaque gène (sachant ses parents) sont alors définies. Par la formule de Bayes, nous cherchons alors la structure de réseau qui maximise la probabilité *a posteriori* (sachant les valeurs observées pour les expressions de gènes). Ces méthodes sont particulièrement efficaces dans l'inférence de réseaux de gènes et leur intérêt majeur est de pouvoir distinguer les interactions directes de celles qui sont indirectes (grâce au conditionnement par rapport aux parents). Ces méthodes, dans lesquelles un algorithme de recherche des réseaux possibles est souvent nécessaire, ne sont pas adaptées aux réseaux contenant plusieurs centaines de gènes. Enfin, nous avons les méthodes basées sur des équations différentielles ou des régressions, dans lesquelles des techniques spécifiques doivent être utilisées, du fait que le nombre d'observations est souvent largement inférieur au nombre de variables (les gènes). Vu sous cet angle, le problème peut se poser sous la forme d'un choix de variables. L'approche la plus courante consiste à pénaliser l'estimation des paramètres dans la régression linéaire, comme dans Gustafsson et al. (2009, 2010). Ces méthodes sont quant à elles particulièrement bien adaptées dans le cadre d'inférence de réseaux de grande taille.

Certaines maladies comme le cancer affectent le programme génique. Il est alors intéressant de chercher à inférer le programme génique des individus sains et des patients malades. Il est légitime de supposer que seule une partie du réseau de gènes soit altérée d'un état à l'autre ; par conséquent, l'estimation simultanée du réseau des individus sains et des patients malades permettrait de prendre en compte l'information commune entre

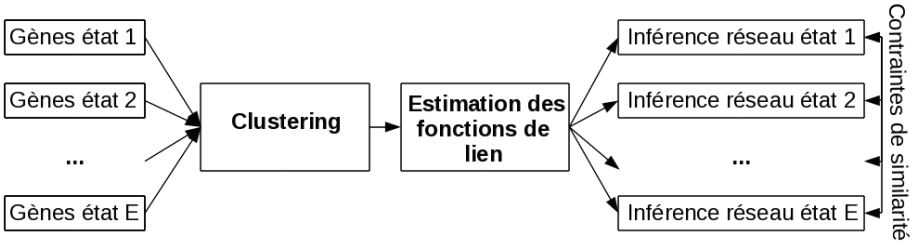


Figure 1: Principe de la méthode proposée pour inférer des réseaux de gènes provenant de plusieurs états

les différents états. Dans les méthodes présentées ci-dessus, seule Dondelinger et al. (2011) permet de prendre en compte cette problématique grâce à un réseau dynamique bayésien, dans le cadre de réseaux de taille limitée. Aussi, nous proposons ici une nouvelle méthode permettant l'estimation simultannée de larges réseaux de gènes issus de multiples états.

Cette méthode, développée ci-dessous, se base sur une régression linéaire précédée d'une étape de clustering ; elle s'inspire d'un premier modèle proposé dans Vallat et al. (in prep) et exposé lors du second colloque international BIO-SI en biostatistique à Rennes. Cette méthode se décompose, comme montré dans la Figure 1, en plusieurs étapes. Un clustering est d'abord réalisé afin de regrouper les gènes ayant une expression similaire au cours du temps. Nous chercherons ensuite des fonctions de liens entre les gènes de différents clusters. Ensuite, nous inférerons les différents réseaux de gènes, en les contraignant à rester similaires dans les parties de réseaux inchangées par la maladie.

2 Etape 1 : Clustering

Supposons que nous disposons d'observations provenant de N gènes, mesurés chacun dans E états, sur T temps de mesure et sur P patients (considérés ici comme des répétitions indépendantes). Ainsi, X_{netp} correspond à l'expression du gène $n \in 1, ..., N$ mesuré pour l'état $e \in 1, ..., E$, au temps $t \in 1, ..., T$ et sur le patient $p \in 1, ..., P$. Nous noterons $\boldsymbol{X}_{net.} = (X_{net1}, ..., X_{netP})'$ le vecteur pour le gène, l'état et le temps fixés ; cette notation est valable quelques soient la place et le nombre de coordonnées remplacées par des points.

Comme montré dans la Figure 1, les expressions de gènes issus des différents états sont mélangées pour le clustering ; autrement dit, notre liste de vecteurs sur laquelle sera appliqué le clustering sera :

$$\{\boldsymbol{X}_{ne..}\}_{n,e}\,.$$

Les deux méthodes les plus courantes pour faire du clustering d'expression de gènes sont les cartes auto-organisatrices (SOMs) et la méthode k-means. Supposons que nous cherchons à classer les gènes en C clusters. Pour prendre en compte les différents états nous pouvons modifier l'algorithme k-means (référence) en cherchant à minimiser la fonction suivante :

$$J(U,V) = \sum_{c=1}^{C} \sum_{e=1}^{E} \sum_{n=1}^{N} \sum_{p=1}^{P} (\alpha_n \mu_{nec} + (1-\alpha_n)\gamma_{nc}) \left(\boldsymbol{X}_{ne.p} - \boldsymbol{V}_c\right)^2 \tag{1}$$

où $U = \{\mu_{nec}, \gamma_{nc}\}_{n,e,c}$ et $V = \{\boldsymbol{V}_c\}_c$, contiennent les profils temporels (vecteurs de longueur T) pour chaque cluster. Nous rajoutons les contraintes $\sum_c \mu_{nec} = \sum_c \gamma_{nc} = 1$. L'ensemble $\{\alpha_n\}_n$ est constitué de constantes fixées *a priori*, comprises entre 0 et 1. Plus α_n est petit, et plus les différents états d'un même gène n seront contraints d'appartenir au même cluster. Une manière de fixer les α_n est de comparer la variabilité de l'expression des patients entre les différentes conditions et à l'intérieur d'une condition donnée :

$$\alpha_n = \max(1 - \alpha'_n, 0)$$

avec :

$$\alpha'_n = \frac{(E-1)\,P}{P-1} \times \frac{\displaystyle\sum_{e=1}^{E} \sum_{p=1}^{P} \sum_{\substack{p'=1 \\ p'>p}}^{P} \left(\boldsymbol{X}_{ne.p} - \boldsymbol{X}_{ne.p'}\right)^2}{\displaystyle\sum_{e=1}^{E} \sum_{\substack{e'=1 \\ e'>e}}^{E} \sum_{p=1}^{P} \sum_{p'=1}^{P} \left(\boldsymbol{X}_{ne.p} - \boldsymbol{X}_{ne'.p'}\right)^2}.$$

Pour minimiser la fonction J de l'équation (1), il faut procéder à un algorithme itératif. L'initialisation, qui détermine l'ensemble V des réprésentants de chaque cluster peut se faire en effectuant une analyse en composantes principales.

Etape 2 : Estimation des fonctions de liens

Nous cherchons maintenant un ensemble de fonctions $\{f_{ij}\}$, $1 \leqslant i, j \leqslant c$, $i \neq j$ qui décrit comment un élement du cluster c_1 (ie., un gène dans un état donné, noté $\boldsymbol{X}_{ne.}$) agit sur un élément du cluster c_2. Nous supposerons que l'état d'un gène au temps t est entièrement régulé par l'état d'autres gènes au temps $t-1$. Les clusters c_i et c_j étant fixés, nous allons chercher à minimiser :

$$\min_{f_{ij}\in\mathcal{F}} \left\{ \sum_{\substack{n_1=1,...,N \\ e_1=1,...,E \\ \boldsymbol{X}_{n_1e_1..}\in c_i}} \sum_{\substack{n_2=1,...,N \\ e_2=1,...,E \\ \boldsymbol{X}_{n_2e_2..}\in c_j}} \sum_{p=1,...,P} \sum_{t=2,...,T} \left\| X_{n_1etp} - f_{ij}\left(X_{n_2e(t-1)p}\right)\right\|_2^2 \right\}$$

avec $\|.\|_2$ la norme euclidienne, et $\mathcal{F}$ un espace de fonction de $\mathbb{R}$ dans $\mathbb{R}$ à définir. Cet espace de fonctions peut être général ou peut contenir un ensemble discret de fonctions choisies *a priori* comme dans Gustafsson et al. (2009).

Etape 3 : Inférer le réseau de gènes

À partir de maintenant, nous décomposons le problème en N problèmes indépendants. Supposons que nous voulons connaître les régulateurs du gène 1 *sachant qu'il appartient aux clusters* $c_1, ..., c_E$ *pour les états respectifs* $1, ..., E$.

Nous transformons d'abord tous les régulateurs potentiels du gène 1, par les fonctions estimées ci-dessus. Précisément :

$$\forall n \in 1, ..., N \quad \forall e \in 1, ..., E \quad \forall t \in 1, ..., T \quad \widetilde{X}_{netp} = f_{cl(n,e)c_e}\left(X_{npet}\right)$$

où $cl(.,.)$ est la fonction qui à un gène et à un état associe son cluster pour l'état en question. Ensuite, nous minimisons :

$$\min_{\{\beta_{n,e}\}\in\mathbb{R}} \left\{ \sum_{e=1}^{E}\sum_{p=1}^{P}\sum_{t=2}^{T} \left\| \boldsymbol{X}_{1etp} - \sum_{n=1}^{N} \beta_{n,e}\widetilde{\boldsymbol{X}}_{ne(t-1)p} \right\|_2^2 + \sum_{e=1}^{E}\sum_{n=1}^{N} \rho(|\beta_{n,e}|\,|\gamma,\lambda_1) \right.$$
$$\left. +\lambda_2 \sum_{e=1}^{E}\sum_{n=1}^{N}\sum_{e'=1}^{E}\sum_{n'=1}^{N} |\theta_{nen'e'}| \left(\beta_{n,e} - sgn\left(\theta_{nen'e'}\right)\beta_{n',e'}\right)^2 \right\}$$

- γ, λ_1 et λ_2 sont des paramètres à estimer par validation croisée,
- $$\rho(t\,|\gamma,\lambda_1) = \lambda_1 \int_0^t \left(1 - \frac{x}{\gamma\lambda_1}\right)_+ dx$$
- $\theta_{nen'e'}$ reflète la proximité *a priori* du paramètre $\beta_{n,e}$ et $\beta_{n',e'}$. Nous pouvons par exemple choisir :
$$\theta_{nen'e'} = \text{corr}\left(\boldsymbol{X}_{ne..}, \boldsymbol{X}_{n'e'..}\right)$$
- $sgn(.)$ est la fonction qui à un nombre associe son signe.

La fonction ρ, présentée dans Zhang (2010), sert à sélectionner les variables dans la régression linéaire. Contrairement à la régression Lasso, dans laquelle tous les termes sont affectés également par la pénalité (introduisant ainsi du biais dans la méthode), les termes supérieurs à $\gamma\lambda_1$ ne seront pas affectés (en effet : $x > \gamma\lambda_1 \Rightarrow \rho'(x) = 0$). La contrainte L_2 a été étudiée dans Huang (2011).

L'ensemble $\{\beta_{n,e} \neq 0\}_n$, pour $e = 1, ..., E$ fixé, réprésente les régulateurs du gène 1 dans l'état e. Précisément, $\beta_{n,e}$ représente l'intensité de l'action du gène n sur le gène 1 dans l'état e.

Bibliographie

[1] Chiquet J. (2011), *Réseaux biologiques*, SMF Gazette, No. 130, 76–82.
[2] Dondelinger F., Husmeier D. et Lèbre S. (2011), *Dynamic Bayesian networks in molecular plant science: inferring gene regulatory networks from multiple gene expression time series*, Euphytica, 1–17.
[3] Gustafsson M., Hornquist M., Lundstrom J., Bjorkegren J., et J. Tegner J. (2009). *Reverse engineering of gene networks with LASSO and nonlinear basis functions*, Annals of the New York Academy of Sciences, Vol. 1158, No. 1, 265–275.
[4] Gustafsson M. et M. Hornquist (2010),*Gene Expression Prediction by Soft Integration and the Elastic Net-Best Performance of the DREAM3 Gene Expression Challenge*, PLoS One, Vol. 5, No. 2, e9134.
[5] Huang J., Ma S., Li H., et Zhang C. (2011), *The sparse Laplacian shrinkage estimator for highdimensional regression*, The Annals of Statistics, Vol. 39, No. 4, 2021–2046.
[6] Liang S., Fuhrman S., et Somogyi R. (1998),*Reveal, a general reverse engineering algorithm for inference of genetic network architectures*, Pacific symposium on biocomputing, Vol. 3, 18–29.
[7] Margolin A., Nemenman I.,. Basso K, Wiggins C.,. Stolovitzky G, Favera R., et Califano A. (2006), *ARACNE : an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context*, BMC Bioinformatics, Vol. 7, S7.
[8] Steinhaus H. (1956), *Sur la division des corps matériels en parties*, Bull. Acad. Polon. Sci., Vol. 1, 801–804.
[9] Vallat L., Kemper C., Jung N., Pocheville A, Maumy-Bertrand M, Bertrand F., Meyer N., Bahram S., Fisher J. et Gribben J. (in prep), *Predicted intervention in a cancer cell genetic program.*
[10] Zhang C. (2010), *Nearly unbiased variable selection under minimax concave penalty*, The Annals of Statistics, Vol. 38, No. 2, 894–942.

Bibliographie

M. Abramowitz et I.A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. National Bureau of Standards Applied Mathematics Series 55. Tenth Printing.* ERIC, 1972. (Cité page 139.)

A. Ahmed et E.P. Xing. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, 106(29) :11878, 2009. (Cité page 27.)

H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6) :716–723, 1974. (Cité page 135.)

T. Akutsu, S. Miyano, et S. Kuhara. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, 16(8) :727, 2000. (Cité page 26.)

R. Albert et A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1) :47, 2002. (Cité page 30.)

U. Alon. Network motifs : theory and experimental approaches. *Nature Reviews Genetics*, 8(6) :450–461, 2007. (Cité pages 108 et 118.)

C. Auffray, Z. Chen, et L. Hood. Systems medicine : the future of medical genomics and healthcare. *Genome Med*, 1(1) :2, 2009. (Cité page 31.)

C. Auffray et L. Nottale. Scale relativity theory and integrative systems biology : 1 : founding principles and scale laws. *Progress in biophysics and molecular biology*, 97(1) :79–114, 2008. (Cité page 31.)

F. Bach. Bolasso : model consistent lasso estimation through the bootstrap. Dans *Proceedings of the 25th international conference on Machine learning*, pages 33–40. ACM, 2008. (Cité pages 59 et 138.)

F. Bach. Model-consistent sparse estimation through the bootstrap. *arXiv preprint arXiv :0901.3202*, 2009. (Cité page 59.)

G. Balazsi, A-L Barabási, et Z. Oltvai. Topological units of environmental signal processing in the transcriptional regulatory network of escherichia coli. *Proceedings of the National Academy of Sciences*, 102(22) :7841–7846, 2005. (Cité page 30.)

M. Bansal, V. Belcastro, A. Ambesi-Impiombato, et D. Di Bernardo. How to infer gene networks from expression profiles. *Molecular systems biology*, 3(1), 2007. (Cité pages 23 et 109.)

M. Bansal, G. Della Gatta, et D. Di Bernardo. Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, 22(7) :815–822, 2006. (Cité page 25.)

Z. Bar-Joseph, A. Gitter, et I. Simon. Studying and modelling dynamic biological processes using time-series gene expression data. *Nature Reviews Genetics*, 13(8) :552–564, 2012. (Cité pages 23, 122 et 167.)

A. Barabási. Emergence of scaling in complex networks. *Handbook of graphs and networks : from the genome to the internet*, pages 69–84, 2002. (Cité pages 123, 124 et 185.)

A. Barabási et R. Albert. Emergence of scaling in random networks. *science*, 286(5439) :509–512, 1999. (Cité pages 21, 22, 29 et 103.)

A. Barabasi et Z. Oltvai. Network biology : understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2) :101–113, 2004. (Cité pages 106 et 108.)

A. Barabási et Z.N. Oltvai. Network biology : understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2) :101–113, 2004. (Cité pages 29 et 167.)

M.J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, et D.L. Wild. A bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21(3) :349–356, 2005. (Cité page 28.)

A. Belloni et V. Chernozhukov. L1-penalized quantile regression in high-dimensional sparse models. *The Annals of Statistics*, 39(1) :82–130, 2011. (Cité page 26.)

C. Bernard. *Introduction à l'étude de la médecine expérimentale*. Baillière, 1865. (Cité page 10.)

G. Beslon et M. Morange. Apprivoiser la vie : Modélisation individu-centrée de systèmes biologiques complexes. *Habilitation à Diriger des Recherches, INSA-Lyon*, 2008. (Cité page 9.)

D. Bhowmick, AC Davison, D.R. Goldstein, et Y. Ruffieux. A laplace mixture model for identification of differential expression in microarray experiments. *Biostatistics*, 7(4) :630, 2006a. (Cité pages 95 et 97.)

D. Bhowmick, AC Davison, D.R. Goldstein, et Y. Ruffieux. A laplace mixture model for identification of differential expression in microarray experiments. *Biostatistics*, 7(4) :630, 2006b. (Cité pages 108 et 115.)

D.R. Bickel, Z. Montazeri, P.C. Hsieh, M. Beatty, S.J. Lawit, et N.J. Bate. Gene network reconstruction from transcriptional dynamics under kinetic model uncertainty : a case for the second derivative. *Bioinformatics*, 25 (6) :772, 2009. (Cité page 26.)

P. Bonacich. Power and centrality : A family of measures. *American journal of sociology*, pages 1170–1182, 1987. (Cité page 17.)

R. Bonneau, D.J. Reiss, P. Shannon, M. Facciotti, L. Hood, N.S. Baliga, et V. Thorsson. The inferelator : an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome biology*, 7(5) :R36, 2006. (Cité pages 26 et 27.)

L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4) :373–384, 1995. (Cité page 36.)

K. Burnham et D. Anderson. *Model selection and multi-model inference : a practical information-theoretic approach*. Springer, 2002. (Cité page 135.)

A. Califano. Rewiring makes the difference. *Molecular systems biology*, 7 (1), 2011. (Cité page 108.)

Emmanuel J Candes, Justin K Romberg, et Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8) :1207–1223, 2006. (Cité page 43.)

W. Cannon. The wisdom of the body. 1932. (Cité page 10.)

A. Cavagna, A. Cimarelli, I. Giardina, G. Parisi, R. Santagati, F. Stefanini, et M. Viale. Scale-free correlations in starling flocks. *Proceedings of the National Academy of Sciences*, 107(26) :11865–11870, 2010. (Cité page 9.)

S. Chao, H. Janping, et J. Sungwon. Inference of gene regulatory networks using time-series data : A survey. *Current Genomics*, 10 :416–429, 2009. (Cité page 23.)

L. Chen, D.B Goldgof, L.O. Hall, et S.A. Eschrich. Noise-based feature perturbation as a selection method for microarray data. Dans *Bioinformatics Research and Applications*, pages 237–247. Springer, 2007. (Cité page 138.)

S. Chen, D. Donoho, et M. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1) :129–159, 2001. (Cité page 135.)

N. Chiorazzi, K. Rai, et M. Ferrarini. Chronic lymphocytic leukemia. *New England Journal of Medicine*, 352(8) :804–815, 2005. (Cité pages 65, 66 et 107.)

J. Chiquet. Reseaux biologiques. *SMF Gazette*, 2011. (Cité page 24.)

S. Christley, Q. Nie, et X. Xie. Incorporating existing network information into gene network inference. *PloS one*, 4(8) :e6799, 2009. (Cité pages 26 et 108.)

A. Clauset, C. Shalizi, et M. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4) :661–703, 2009. (Cité pages 123, 129 et 182.)

G. Clermont, C. Auffray, Y. Moreau, D. Rocke, D. Dalevi, D. Dubhashi, D. R Marshall, P. Raasch, F. Dehne, P. Provero, *et al.* Bridging the gap between systems biology and medicine. *Genome Med*, 1(9) :88, 2009. (Cité page 31.)

Collectif. Larousse. *Dictionnaire, Paris, Larousse*, 2008. (Cité page 7.)

J.R. Cook et L.A. Stefanski. Simulation-extrapolation estimation in parametric measurement error models. *Journal of the American Statistical Association*, 89(428) :1314–1328, 1994. (Cité page 138.)

G. M Cooper, JA Johnson, TY. Langaee, H. Feng, I. B Stanaway, U. Schwarz, M. Ritchie, M. Stein, Dan M. Roden, J. Smith, *et al.* A genome-wide scan for common genetic variants with a large influence on warfarin maintenance dose. *Blood*, 112(4) :1022–1027, 2008. (Cité page 32.)

L. Cope, R. Irizarry, H. Jaffee, Z. Wu, et T. Speed. A benchmark for affymetrix genechip expression measures. *Bioinformatics*, 20(3) :323–331, 2004. (Cité page 152.)

F. Crick *et al.* Central dogma of molecular biology. *Nature*, 227(5258) : 561–563, 1970. (Cité pages 12 et 167.)

G. Csardi et T. Nepusz. The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5), 2006. (Cité page 180.)

B. Da Wei Huang et R. Lempicki. Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature protocols*, 4 (1) :44–57, 2008. (Cité page 71.)

R. Damle, T. Wasil, F. Fais, F. Ghiotto, A. Valetto, S. Allen, A. Buchbinder, D. Budman, K. Dittmar, J. Kolitz, *et al.* Ig v gene mutation status and cd38 expression as novel prognostic indicators in chronic lymphocytic leukemia presented in part at the 40th annual meeting of the american society of hematology, held in miami beach, fl, december 4-8, 1998. *Blood*, 94(6) :1840–1847, 1999. (Cité page 66.)

R. De Angelis, M. Sant, M. Coleman, S. Francisci, P. Baili, D. Pierannunzio, A. Trama, O. Visser, H. Brenner, E. Ardanaz, *et al.* Cancer survival in europe 1999–2007 by country and age : results of eurocare-5—a population-based study. *The lancet oncology*, 15(1) :23–34, 2014. (Cité page 1.)

H. De Jong. Modeling and simulation of genetic regulatory systems : a literature review. *Journal of computational biology*, 9(1) :67–103, 2002. (Cité pages 22 et 23.)

H. den Ham, L. Waal, F. Zaaraoui-Boutahar, M. Bijl, W. IJcken, A. Osterhaus, R. Boer, et A. Andeweg. Early divergence of th1 and th2 transcriptomes involves a small core response and sets of transiently expressed genes. *European journal of immunology*, 43(4) :1074–1084, 2013. (Cité pages 121, 124, 127, 191, 193 et 196.)

A. Deplancke, B.and Mukhopadhyay, W. Ao, A. Elewa, C. Grove, N. Martinez, R. Sequerra, L. Doucette-Stamm, J. Reece-Hoyes, I. Hope, *et al.* A gene-centered< i> c. elegans</i> protein-dna interaction network. *Cell*, 125(6) :1193–1205, 2006. (Cité page 29.)

B. Di Camillo, B.A. Irving, J. Schimke, T. Sanavia, G. Toffolo, C. Cobelli, et K.S. Nair. Function-based discovery of significant transcriptional temporal patterns in insulin stimulated muscle cells. *PloS one*, 7(3) :e32391, 2012. (Cité pages 108 et 201.)

D. Donoho et M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, 100(5) :2197–2202, 2003. (Cité pages 43 et 135.)

D. Donoho, M. Elad, et V. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *Information Theory, IEEE Transactions on*, 52(1) :6–18, 2006. (Cité page 57.)

D. Donoho *et al.* High-dimensional data analysis : The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 2000. (Cité page 134.)

C. Eckart et G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3) :211–218, 1936. (Cité page 25.)

B. Efron. *The jackknife, the bootstrap and other resampling plans*, volume 38. SIAM, 1982. (Cité page 59.)

B. Efron, T. Hastie, I. Johnstone, et R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2) :407–499, 2004. (Cité pages 26, 35, 41, 136 et 145.)

M. Eklund et S. Zwanzig. Simsel : a new simulation method for variable selection. *Journal of Statistical Computation and Simulation*, 82(4) :515–527, 2012. (Cité page 138.)

P. Erdos et A. Renyi. On random graphs i. *Publ. Math. Debrecen*, 6 : 290–297, 1959. (Cité page 18.)

J. Ernst, G.J. Nau, et Z. Bar-Joseph. Clustering short time series gene expression data. *Bioinformatics*, 21(suppl 1) :i159, 2005. ISSN 1367-4803. (Cité page 116.)

J. Fan. Comments on «wavelets in statistics : A review» by a. antoniadis. *Journal of the Italian Statistical Society*, 6(2) :131–138, 1997. (Cité page 136.)

J. Fan et R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96 (456) :1348–1360, 2001. (Cité page 134.)

J. Fan et R. Li. Statistical challenges with high dimensionality : Feature selection in knowledge discovery. *arXiv preprint math/0602133*, 2006. (Cité pages 133, 134 et 135.)

J. Fan et J. Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1) :101, 2010. (Cité page 135.)

E. Foster, D.and George. The risk inflation criterion for multiple regression. *The Annals of Statistics*, pages 1947–1975, 1994. (Cité page 135.)

L. E. Frank et J.H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2) :109–135, 1993. (Cité page 136.)

H. Fraser, A. Hirsh, L. Steinmetz, C. Scharfe, et M. Feldman. Evolutionary rate in the protein interaction network. *Science*, 296(5568) :750–752, 2002. (Cité page 29.)

J. Friedman, T. Hastie, et R. Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv :1001.0736*, 2010. (Cité pages 59 et 136.)

N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659) :799, 2004. (Cité page 23.)

N. Friedman, M. Linial, I. Nachman, et D. Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4) : 601–620, 2000. (Cité page 28.)

T. Gaikwad, K. Ghosh, B. Kulkarni, V. Kulkarni, C. Ross, et S. Shetty. Influence of cyp2c9 and vkorc1 gene polymorphisms on warfarin dosage, over anticoagulation and other adverse outcomes in indian population. *European journal of pharmacology*, 710(1) :80–84, 2013. (Cité page 32.)

T.S. Gardner, D. di Bernardo, D. Lorenz, et J.J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629) :102, 2003. (Cité pages 25 et 107.)

P. Ghia, A. Ferreri, et F. Caligaris-Cappio. Chronic lymphocytic leukemia. *Critical reviews in oncology/hematology*, 64(3) :234–246, 2007. (Cité page 65.)

G. Giaever, L. Chu, A.and Ni, C. Connelly, L. Riles, S. Véronneau, S. Dow, A. Lucau-Danila, K. Anderson, B. André, *et al.* Functional profiling of the saccharomyces cerevisiae genome. *Nature*, 418(6896) :387–391, 2002. (Cité page 30.)

A. Giri, N. Khan, S. Grover, I. Kaur, A. Basu, N. Tandon, V. Scaria, IGV Consortium, R. Kukreti, S. Brahmachari, *et al.* Genetic epidemiology of pharmacogenetic variations in cyp2c9, cyp4f2 and vkorc1 genes associated with warfarin dosage in the indian population. *Pharmacogenomics*, 15 (10) :1337–1354, 2014. (Cité page 32.)

N. Goldenfeld et L. Kadanoff. Simple lessons from complexity. *Science*, 284 (5411) :87–89, 1999. (Cité page 8.)

X. Gourdon. *Les maths en tête : analyse : mathématiques pour M'*. Ellipses, 2000. (Cité page 43.)

F. Gregoretti, V. Belcastro, D. di Bernardo, et G. Oliva. A parallel implementation of the network identification by multiple regression (nir) algorithm to reverse-engineer regulatory gene networks. *PLoS One*, 5(4) : e10179, 2010. (Cité page 25.)

A. Guarini, S. Chiaretti, S. Tavolaro, R. Maggio, N. Peragine, F. Citarella, M.R. Ricciardi, S. Santangelo, M. Marinelli, M.S. De Propris, *et al.* Bcr ligation induced by igm stimulation results in gene expression and functional changes only in igvh unmutated chronic lymphocytic leukemia (cll) cells. *Blood*, 112(3) :782–792, 2008. (Cité page 107.)

Z. Guo, M. Maki, R. Ding, Y. Yang, L. Xiong, *et al.* Genome-wide survey of tissue-specific microrna and transcription factor regulatory networks in 12 tissues. *Scientific Reports*, 4, 2014. (Cité page 13.)

M. Gustafsson et M. Hörnquist. Gene expression prediction by soft integration and the elastic net-best performance of the dream3 gene expression challenge. *PLoS One*, 5(2) :e9134, 2010. (Cité page 26.)

M. Gustafsson, M. Hörnquist, J. Lundström, J. Björkegren, et J. Tegnér. Reverse engineering of gene networks with lasso and nonlinear basis functions. *Annals of the New York Academy of Sciences*, 1158(1) :265–275, 2009. (Cité page 27.)

T. Hamblin, Z. Davis, A. Gardiner, D. Oscier, et F. Stevenson. Unmutated ig vh genes are associated with a more aggressive form of chronic lymphocytic leukemia. *Blood*, 94(6) :1848–1854, 1999. (Cité pages 66 et 107.)

J.-D.J Han, N. Bertin, T. Hao, D. Goldberg, G. Berriz, L. Zhang, D. Dupuy, A. Walhout, M. Cusick, F. Roth, *et al.* Evidence for dynamically organized modularity in the yeast protein–protein interaction network. *Nature*, 430 (6995) :88–93, 2004. (Cité page 30.)

S. Hao et D. Baltimore. The stability of mrna influences the temporal order of the induction of genes encoding inflammatory molecules. *Nature immunology*, 10(3) :281–288, 2009. (Cité pages 108 et 123.)

J. Hasty, D. McMillen, et J. Collins. Engineered gene circuits. *Nature*, 420 (6912) :224–230, 2002. (Cité page 10.)

F. He, R. Balling, et A.P. Zeng. Reverse engineering and verification of gene networks : principles, assumptions, and limitations of present methods and future perspectives. *Journal of biotechnology*, 144(3) :190–203, 2009. (Cité page 23.)

M. Hecker, S. Lambeck, S. Toepfer, E. Van Someren, et R. Guthke. Gene regulatory network inference : data integration in dynamic models—a review. *Biosystems*, 96(1) :86–103, 2009. (Cité pages 23, 106, 107, 122 et 167.)

Y. Herishanu, P. Pérez-Galán, D. Liu, A. Biancotto, S. Pittaluga, B. Vire, F. Gibellini, N. Njuguna, E. Lee, L. Stennett, *et al.* The lymph node microenvironment promotes b-cell receptor signaling, nf-κb activation, and tumor proliferation in chronic lymphocytic leukemia. *Blood*, 117(2) :563–574, 2011. (Cité page 107.)

R. R. Hocking. A biometrics invited paper. the analysis and selection of variables in linear regression. *Biometrics*, 32(1) :1–49, 1976. (Cité page 135.)

A. Hoerl et R. Kennard. Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics*, 12(1) :55–67, 1970. (Cité page 135.)

L. Hood, R. Balling, et C. Auffray. Revolutionizing medicine in the 21st century through systems approaches. *Biotechnology journal*, 7(8) :992–1001, 2012. (Cité page 31.)

J. Huang, S. Ma, H. Li, et C.H. Zhang. The sparse laplacian shrinkage estimator for high-dimensional regression. *The Annals of Statistics*, 39 (4) :2021–2046, 2011. (Cité page 26.)

T. Huang, L. Liu, Z. Qian, K. Tu, Y. Li, et L. Xie. Using genereg to construct time delay gene regulatory networks. *BMC research notes*, 3(1) :142, 2010. (Cité pages 101, 109, 118, 161, 162 et 164.)

D. Husmeier. Introduction to learning bayesian networks from data. *Probabilistic modeling in bioinformatics and medical informatics*, pages 17–57, 2005. (Cité page 27.)

T. Ideker, L. Winslow, et D. Lauffenburger. Bioengineering and systems biology. *Annals of biomedical engineering*, 34(7) :1226–1233, 2006. (Cité pages xiii, 11 et 12.)

R. Irizarry, B. Hobbs, F. Collin, Y. Beazer-Barclay, K. Antonellis, U. Scherf, et T. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2) :249–264, 2003. (Cité page 152.)

J. Ivanic, X. Yu, A. Wallqvist, et J. Reifman. Influence of protein abundance on high-throughput protein-protein interaction detection. *PloS one*, 4(6) : e5815, 2009. (Cité page 29.)

W. James et C. Stein. Estimation with quadratic loss. Dans *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability : held at the Statistical Laboratory, University of California, June 20-July 30, 1960*, page 361. Univ of California Press, 1961. (Cité page 25.)

I. Jang, A. Margolin, et A. Califano. haracne : improving the accuracy of regulatory model reverse engineering via higher-order data processing inequality tests. *Interface focus*, 3(4) :20130011, 2013. (Cité page 24.)

A. Järvinen, S. Hautaniemi, H. Edgren, P. Auvinen, J. Saarela, O. Kallioniemi, et O. Monni. Are data from different gene expression microarray platforms comparable ? *Genomics*, 83(6) :1164–1168, 2004. (Cité page 152.)

H. Jeong, S. Mason, A-L Barabási, et Z. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833) :41–42, 2001. (Cité pages 29 et 30.)

H. Jeong, Z. Néda, et A. Barabási. Measuring preferential attachment in evolving networks. *EPL (Europhysics Letters)*, 61(4) :567, 2003. (Cité page 185.)

H. Jeong, B. Tombor, R. Albert, Z. Oltvai, et A-L Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804) :651–654, 2000. (Cité pages 22 et 182.)

P. Jonsson et P. Bates. Global topological features of cancer proteins in the human interactome. *Bioinformatics*, 22(18) :2291–2297, 2006. (Cité page 30.)

N. Jung, F. Bertrand, S. Bahram, L. Vallat, et M. Maumy-Bertrand. Cascade : a r package to study, predict and simulate the diffusion of a signal through a temporal gene network. *Bioinformatics*, 30(4) :571–573, 2014. (Cité pages 3, 121 et 156.)

W. Karush. *Minima of functions of several variables with inequalities as side conditions*. PhD thesis, University of Chicago, Department of Mathematics., 1939. (Cité page 43.)

CA. Kemper. *Exploiting biological pathways to infer temporal gene interaction models*. PhD thesis, Massachusetts Institute of Technology, 2006. (Cité pages 79 et 91.)

S. Kim, S. Imoto, et S. Miyano. Dynamic bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems*, 75(1-3) :57–65, 2004. (Cité page 27.)

H. Kishino, P.J. Waddell, *et al.* Correspondence analysis of genes and tissue types and finding genetic links from microarray data. *GENOME INFORMATICS SERIES*, pages 83–95, 2000. (Cité page 24.)

H. Kitano. Perspectives on systems biology. *New generation Computing*, 18 (3) :199–216, 2000. (Cité page 7.)

H. Kitano. 1 systems biology : Toward system-level understanding of biological systems. *Foundations of systems biology*, page 1, 2001. (Cité page 7.)

H. Kitano. Computational systems biology. *Nature*, 420(6912) :206–210, 2002a. (Cité page 10.)

H. Kitano. Systems biology : a brief overview. *Science*, 295(5560) :1662–1664, 2002b. (Cité page 10.)

H. Kitano. Systems biology : a brief overview. *Science*, 295(5560) :1662, 2002c. ISSN 0036-8075. (Cité page 106.)

K. Knight et W. Fu. Asymptotics for lasso-type estimators. *Annals of Statistics*, pages 1356–1378, 2000. (Cité pages 52 et 59.)

J. R. Koza, F. Bennett III, et O. Stiffelman. *Genetic programming as a Darwinian invention machine*. Springer, 1999. (Cité page 135.)

K.-A. Lê Cao, S. Boitard, et P. Besse. Sparse pls discriminant analysis : biologically relevant feature selection and graphical displays for multiclass problems. *BMC bioinformatics*, 12(1) :253, 2011. (Cité page 70.)

K.A. Lê Cao, D. Rossouw, C. Robert-Granié, et P. Besse. A sparse pls for variable selection when integrating omics data. *Statistical applications in genetics and molecular biology*, 7(1) :35, 2008. (Cité page 26.)

R. Leclerc. Survival of the sparsest : robust gene networks are parsimonious. *Molecular systems biology*, 4(1), 2008. (Cité page 23.)

T.I. Lee, N.J. Rinaldi, F. Robert, D.T. Odom, Z. Bar-Joseph, G.K. Gerber, N.M. Hannett, C.T. Harbison, C.M. Thompson, I. Simon, *et al.* Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, 298 (5594) :799, 2002. ISSN 0036-8075. (Cité page 106.)

H. Leeb et B.M. Pötscher. Sparse estimators and the oracle property, or the return of hodges' estimator. *Journal of Econometrics*, 142(1) :201–211, 2008. (Cité page 26.)

C. Li et W. H. Wong. Dna-chip analyzer (dchip). Dans *The Analysis of Gene Expression Data*, pages 120–141. Springer, 2003. (Cité page 152.)

C. Li et W.H. Wong. Model-based analysis of oligonucleotide arrays : expression index computation and outlier detection. *Proceedings of the National Academy of Sciences of the United States of America*, 98(1) :31, 2001. (Cité pages 110 et 119.)

R. J. Lipshutz, S. PA. Fodor, T. R Gingeras, et D. J. Lockhart. High density synthetic oligonucleotide arrays. *Nature genetics*, 21 :20–24, 1999. (Cité page 134.)

Y.Y. Liu, J.J. Slotine, et A.L. Barabási. Controllability of complex networks. *Nature*, 473(7346) :167–173, 2011. (Cité page 106.)

J. Long et M. Roth. Synthetic microarray data generation with range and nemo. *Bioinformatics*, 24(1) :132–134, 2008. (Cité pages 103, 108 et 118.)

X. Luo, L.A. Stefanski, et D.D. Boos. Tuning variable selection procedures by adding noise. *Technometrics*, 48(2) :165–175, 2006. (Cité page 138.)

N.M. Luscombe, M.M. Babu, H. Yu, M. Snyder, S.A. Teichmann, et M. Gerstein. Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, 431(7006) :308–312, 2004. ISSN 0028-0836. (Cité pages 106, 108, 122 et 167.)

C. L. Mallows. Some comments on cp. *Technometrics*, 15(4) :661–675, 1973. (Cité page 135.)

D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, M. Kellis, J. J. Collins, et G. Stolovitzky. Wisdom of crowds for robust gene network inference. *Nature methods*, 2012. (Cité page 102.)

D. Marbach, R. Prill, T. Schaffter, C. Mattiussi, D. Floreano, et G. Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences*, 107(14) : 6286–6291, 2010. (Cité page 106.)

A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Favera, et A. Califano. Aracne : an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics*, 7 (Suppl 1) :S7, 2006a. (Cité pages 24, 118 et 161.)

A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Favera, et A. Califano. Aracne : an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics*, 7 (Suppl 1) :S7, 2006b. (Cité page 101.)

F. Markowetz et R. Spang. Inferring cellular networks- a review. *BMC bioinformatics*, 8(Suppl 6) :S5, 2007. (Cité page 23.)

N. Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1) :374–393, 2007. (Cité page 57.)

N. Meinshausen et P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3) :1436–1462, 2006. (Cité page 54.)

N. Meinshausen et P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 72(4) :417–473, 2010. (Cité pages 138 et 153.)

B.T. Messmer, E. Albesiano, D.G. Efremov, F. Ghiotto, S.L. Allen, J. Kolitz, R. Foa, R.N. Damle, F. Fais, D. Messmer, *et al.* Multiple distinct sets of stereotyped antigen receptors indicate a role for antigen in promoting chronic lymphocytic leukemia. *The Journal of experimental medicine*, 200 (4) :519, 2004. ISSN 0022-1007. (Cité page 107.)

P.W. Mielke et K.J. Berry. *Permutation methods : a distance function approach*. Springer Verlag, 2007. ISBN 0387698116. (Cité page 116.)

C. Mitchell, N. Gregersen, et A. Krause. Novel cyp2c9 and vkorc1 gene variants associated with warfarin dosage variability in the south african black population. *Pharmacogenomics*, 12(7) :953–963, 2011. (Cité page 32.)

E. Morin. *La méthode : la nature de la nature*. Seuil, 2013. (Cité page 9.)

E.R. Morrissey, M.A. Juárez, K. J Denby, et N.J. Burroughs. Inferring the time-invariant topology of a nonlinear sparse gene regulatory network using fully bayesian spline autoregression. *Biostatistics*, 12(4) :682–694, 2011. (Cité pages 101, 109, 118, 161 et 162.)

K. Murphy et S. Mian. Modelling gene expression data using dynamic bayesian networks. Rapport technique, Technical report, Computer Science Division, University of California, Berkeley, CA, 1999. (Cité page 28.)

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2) :227–234, 1995. (Cité page 135.)

M. Newman. Scientific collaboration networks. i. network construction and fundamental results. *Physical review E*, 64(1) :016131, 2001. (Cité page 22.)

L. Nottale et C. Auffray. Scale relativity theory and integrative systems biology : 2 macroscopic quantum-type mechanics. *Progress in biophysics and molecular biology*, 97(1) :115–157, 2008. (Cité page 31.)

R. Opgen-Rhein et K. Strimmer. Learning causal networks from systems biology time course data : an effective model selection procedure for the vector autoregressive process. *BMC bioinformatics*, 8(Suppl 2) :S3, 2007. (Cité page 25.)

T. Opsahl, F. Agneessens, et J. Skvoretz. Node centrality in weighted networks : Generalizing degree and shortest paths. *Social Networks*, 32(3) : 245–251, 2010. (Cité pages 130 et 184.)

J. A Papin et B. Palsson. Topological analysis of mass-balanced signaling networks : a framework to obtain network properties including crosstalk. *Journal of theoretical biology*, 227(2) :283–297, 2004. (Cité page 30.)

R. Pastor-Satorras, E. Smith, et R. Solé. Evolving protein interaction networks through gene duplication. *Journal of Theoretical biology*, 222(2) : 199–210, 2003. (Cité page 29.)

R. Penrose. A generalized inverse for matrices. Dans *Proc. Cambridge Philos. Soc*, volume 51, page C655. Cambridge Univ Press, 1955. (Cité page 25.)

M. Pérez-Enciso et M. Tenenhaus. Prediction of clinical outcome with microarray data : a partial least squares discriminant analysis (pls-da) approach. *Human genetics*, 112(5-6) :581–592, 2003. (Cité page 70.)

A. Perrot, C. Pionneau, S. Nadaud, F. Davi, V. Leblond, F. Jacob, H. Merle-Béral, R. Herbrecht, M.C. Béné, J.G. Gribben, *et al.* A unique proteomic profile on surface igm ligation in unmutated chronic lymphocytic leukemia. *Blood*, 118(4) :e1–e15, 2011. (Cité page 107.)

H. Poincaré. La science et l'hypothèse. *Science*, 6 :1–13, 1898. (Cité page 8.)

D. Price. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, 27 (5) :292–306, 1976. (Cité page 21.)

M. Quach, N. Brunel, et F. d'Alché Buc. Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics*, 23(23) :3209, 2007. (Cité page 28.)

J. Quackenbush. Microarray data normalization and transformation. *Nature genetics*, 32 :496–501, 2002. (Cité page 152.)

L. Rassenti, L. Huynh, T. L Toy, L. Chen, M. Keating, J. Gribben, D. Neuberg, I. Flinn, K. Rai, J. Byrd, *et al.* Zap-70 compared with immunoglobulin heavy-chain gene mutation status as a predictor of disease progression in chronic lymphocytic leukemia. *New England Journal of Medicine*, 351 (9) :893–901, 2004. (Cité page 66.)

A. Rau. *Reverse engineering gene networks using genomic time-course data.* PhD thesis, PURDUE UNIVERSITY, 2011. (Cité page 27.)

A. Rau, F. Jaffrézic, J.L. Foulley, et R.W. Doerge. An empirical bayesian method for estimating biological networks from temporal microarray data. *Statistical Applications in Genetics and Molecular Biology*, 9(1) :9, 2010. (Cité pages 28 et 157.)

S. Rogers et M. Girolami. A bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics*, 21(14) : 3131, 2005. (Cité page 26.)

E. Rozman, C.and Montserrat. Chronic lymphocytic leukemia. *New England Journal of Medicine*, 333(16) :1052–1057, 1995. (Cité page 66.)

J. Ruan. A top-performing algorithm for the dream3 gene expression prediction challenge. *PloS one*, 5(2) :e8944, 2010. (Cité pages 24 et 25.)

M. Said, T. Begley, A. Oppenheim, D. Lauffenburger, et L. Samson. Global network analysis of phenotypic effects : protein networks and toxicity modulation in saccharomyces cerevisiae. *Proceedings of the National Academy of Sciences*, 101(52) :18006–18011, 2004. (Cité page 30.)

J. Schafer et K. Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6) :754, 2005. ISSN 1367-4803. (Cité pages 24, 25, 101, 106, 109, 110, 118, 161, 162 et 164.)

J. Schafer et K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4(1) :32, 2005. (Cité page 25.)

R. Schroers, F. Griesinger, L. Trümper, D. Haase, B. Kulle, L. Klein-Hitpass, L. Sellmann, U. Dührsen, et J. Dürig. Combined analysis of zap-70 and cd38 expression as a predictor of disease progression in b-cell chronic lymphocytic leukemia. *Leukemia*, 19(5) :750–758, 2005. (Cité page 66.)

G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2) :461–464, 1978. (Cité page 135.)

J. Seebacher et A. Gavin. Snapshot : Protein-protein interaction networks. *Cell*, 144(6) :1000–1000, 2011. (Cité pages 29 et 30.)

M. Segal, K. Dahlquist, et B.R. Conklin. Regression approaches for microarray data analysis. *Journal of Computational Biology*, 10(6) :961–980, 2003. (Cité page 134.)

S. Shen-Orr, R. Milo, S. Mangan, et U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics*, 31(1) : 64–68, 2002. (Cité page 30.)

B.T. Sherman, R.A. Lempicki, *et al.* Bioinformatics enrichment tools : paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1) :1–13, 2009. (Cité pages 110 et 111.)

G. Smyth. Limma : linear models for microarray data. Dans R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, et W. Huber, éditeurs, *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, pages 397–420. Springer, New York, 2005. (Cité pages 68, 123, 127 et 171.)

P. Sobradillo, F. Pozo, et Á. Agustí. P4 medicine : the future around the corner. *Archivos de Bronconeumología ((English Edition))*, 47(1) :35–40, 2011. (Cité page 31.)

S. Sra. A short note on parameter approximation for von mises-fisher distributions : and a fast implementation of i s (x). *Computational Statistics*, 27(1) :177–190, 2012. (Cité page 139.)

F. Stevenson et F. Caligaris-Cappio. Chronic lymphocytic leukemia : revelations from the b-cell receptor. *Blood*, 103(12) :4389–4395, 2004. (Cité pages 66 et 107.)

J. Sun, X. Gong, B. Purow, et Z. Zhao. Uncovering microrna and transcription factor mediated regulatory networks in glioblastoma. *PLoS computational biology*, 8(7) :e1002488, 2012. (Cité page 13.)

F. Takeuchi, R. McGinnis, S. Bourgeois, C. Barnes, N. Eriksson, N. Soranzo, P. Whittaker, V. Ranganath, V. Kumanduri, W. McLaren, *et al.* A genome-wide association study confirms vkorc1, cyp2c9, and cyp4f2 as principal genetic determinants of warfarin dose. *PLoS genetics*, 5(3) : e1000433, 2009. (Cité page 32.)

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. (Cité pages 26, 35, 37, 41, 135, 136, 152 et 177.)

R. Tibshirani. Regression shrinkage and selection via the lasso : a retrospective. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 73(3) :273–282, 2011. (Cité page 36.)

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, et K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(1) :91–108, 2005. (Cité page 58.)

R.J. Tibshirani. The lasso problem and uniqueness. *arXiv preprint arXiv :1206.0313*, 2012. (Cité page 47.)

L. Vallat, C. Kemper, N. Jung, M. Maumy-Bertrand, F. Bertrand, N. Meyer, A. Pocheville, J. Fisher, J. Gribben, et S. Bahram. Reverse-engineering the genetic circuitry of a cancer cell with predicted intervention in chronic lymphocytic leukemia. *Proceedings of the National Academy of Sciences*, 110(2) :459–464, 2013. (Cité pages 3, 105, 122, 123, 124, 155, 156, 167, 178, 179 et 196.)

L.D. Vallat, Y. Park, C. Li, et J.G. Gribben. Temporal genetic program following B-cell receptor cross-linking : altered balance between proliferation and death in healthy and malignant B cells. *Blood*, 109(9) :3989, 2007. (Cité pages 66, 107, 110, 112, 118, 119, 124 et 169.)

C.J. Van Rijsbergen. Information retrieval. dept. of computer science, university of glasgow. *URL : citeseer. ist. psu. edu/vanrijsbergen79information. html*, 1979. (Cité pages 109 et 123.)

A. Vazquez, R. Pastor-Satorras, et A. Vespignani. Internet topology at the router and autonomous system level. *arXiv preprint cond-mat/0206084*, 2002. (Cité page 22.)

N. Verzelen *et al.* Minimax risks for sparse regressions : Ultra-high dimensional phenomenons. *Electronic Journal of Statistics*, 6 :38–90, 2012. (Cité page 56.)

M. Vidal, M. Cusick, et A.-L. Barabasi. Interactome networks and human disease. *Cell*, 144(6) :986–998, 2011. (Cité page 29.)

Al. Villaverde, J. Ross, et J. Banga. Reverse engineering cellular networks with information theoretic methods. *Cells*, 2(2) :306–329, 2013. (Cité pages xiv, 24 et 33.)

B. Vogelstein et K. Kinzler. Cancer genes and the pathways they control. *Nature medicine*, 10(8) :789–799, 2004. (Cité page 65.)

B. Vogelstein, D. Lane, et A. Levine. Surfing the p53 network. *Nature*, 408 (6810) :307–310, 2000. (Cité page 30.)

M. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (lasso). *Information Theory, IEEE Transactions on*, 55(5) :2183–2202, 2009. (Cité pages 55 et 56.)

M. Walhout, M. Vidal, et J. Dekker. *Handbook of systems biology : concepts and insights*. Academic Press, 2012. (Cité page 13.)

S. Wang, B. Nan, S. Rosset, et J. Zhu. Random lasso. *The annals of applied statistics*, 5(1) :468, 2011. (Cité page 138.)

D. Watts et S. Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684) :440–442, 1998. (Cité pages 16 et 20.)

D.C. Weaver, C.T. Workman, G.D. Stormo, *et al.* Modeling regulatory networks with weight matrices. Dans *Pacific symposium on biocomputing*, volume 4, pages 112–123, 1999. (Cité pages 108 et 118.)

G. Whitesides et R. Ismagilov. Complexity in chemistry. *science*, 284(5411) : 89–92, 1999. (Cité pages 8 et 9.)

H. Wold. Partial least squares. *Encyclopedia of statistical sciences*, 1985. (Cité page 70.)

O. Wolkenhauer, C. Auffray, R. Jaster, G. Steinhoff, et O. Dammann. The road from systems biology to systems medicine. *Pediatric research*, 73 (4-2) :502–507, 2013. (Cité page 31.)

D.D. Wu, Y.and Boos et L.A. Stefanski. Controlling variable selection by the addition of pseudovariables. *Journal of the American Statistical Association*, 102(477), 2007. (Cité page 137.)

T. T. Wu et K. Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, pages 224–244, 2008. (Cité page 136.)

Z. Wu et R. Irizarry. Preprocessing of oligonucleotide array data. *Nature biotechnology*, 22(6) :656–658, 2004. (Cité page 152.)

R. Xu, G.K. Venayagamoorthy, et D.C. Wunsch II. Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. *Neural Networks*, 20(8) :917–927, 2007. (Cité page 27.)

H. K. Yalamanchili, B. Yan, M. Li, J. Qin, Z. Zhao, F. Chin, et J. Wang. Ddgni : Dynamic delay gene-network inference from high-temporal data using gapped local alignment. *Bioinformatics*, 30(3) :377–383, 2014. (Cité page 24.)

MK Yeung, J. Tegnér, et J.J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences*, 99(9) :6163, 2002. (Cité page 25.)

N. Yosef et A. Regev. Impulse control : temporal dynamics in gene transcription. *Cell*, 144(6) :886–896, 2011. (Cité pages 106, 108, 118, 122, 167 et 173.)

W. Young, A. Raftery, et K. Yeung. Fast bayesian inference for gene regulatory networks using scanbma. *BMC systems biology*, 8(1) :47, 2014. (Cité page 28.)

H. Yu, P. Braun, M. Yıldırım, I. Lemmens, K. Venkatesan, J. Sahalie, T. Hirozane-Kishikawa, F. Gebreab, N. Li, N. Simonis, *et al.* High-quality binary protein interaction map of the yeast interactome network. *Science*, 322(5898) :104–110, 2008. (Cité page 30.)

J. Yu, V.A. Smith, P.P. Wang, A.J. Hartemink, et E.D. Jarvis. Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18) :3594–3603, 2004. (Cité page 28.)

M. Yuan et Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 68(1) :49–67, 2006. (Cité pages 58, 59 et 136.)

T. Zenz, D. Mertens, R. Küppers, et S. Döhner, H.and Stilgenbauer. From pathogenesis to treatment of chronic lymphocytic leukaemia. *Nature Reviews Cancer*, 10(1) :37–50, 2009. (Cité page 66.)

C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2) :894–942, 2010. (Cité pages 26 et 136.)

P. Zhao et B. Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7 :2541–2563, 2006. (Cité pages 53, 54 et 136.)

X. Zhu, M. Gerstein, et M. Snyder. Getting connected : analysis and principles of biological networks. *Genes & development*, 21(9) :1010–1024, 2007. (Cité pages 122 et 167.)

P. Zoppoli, S. Morganella, et M. Ceccarelli. Timedelay-aracne : Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC bioinformatics*, 11(1) :154, 2010. (Cité pages 24, 101, 107, 108, 109, 110, 118, 161, 162 et 164.)

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476) :1418–1429, 2006. (Cité pages 57 et 136.)

H. Zou et T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(2) :301–320, 2005. (Cité pages 26, 59, 136 et 138.)

UNIVERSITÉ DE STRASBOURG

Nicolas JUNG

Modélisation de phénomènes biologiques complexes : application à l'étude de la réponse antigénique de lymphocytes B sains et tumoraux.

Résumé

La biologie des systèmes complexes est le cadre idéal pour l'interdisciplinarité. Dans cette thèse, les modèles et les théories statistiques répondent aux modèles et aux expérimentations biologiques.

Nous nous sommes intéressés au cas particulier de la leucémie lymphoïde chronique à cellules B, qui est une forme de cancer des cellules du sang. Nous avons commencé par modéliser le programme génique tumoral sous-jacent à cette maladie et nous l'avons comparé au programme génique d'individus sains. Pour ce faire, nous avons introduit la notion de réseau en cascade. Nous avons ensuite démontré notre capacité à contrôler ce système complexe, en prédisant mathématiquement les effets d'une expérience d'intervention consistant à inhiber l'expression d'un gène. Cette thèse s'achève sur la perspective d'une modulation orientée, c'est-à-dire le choix d'expériences d'intervention permettant de « reprogrammer » le programme génique tumoral vers un état normal.

Réseaux de gènes – Régression Lasso – Biologie des systèmes complexes

Summary

System biology is a well-suited context for interdisciplinary. In this thesis, statistical models and theories closely meet biological models and experiments.

We focused on a specific complex system model: the chronic B-cell chronic lymphocytic leukemia disease which is a cancer of the blood cells. We started by modeling the genetic program which underlies this disease and we compared it to the healthy one. This conduced us to introduce the concept of cascade networks. We then showed our ability to control this complex system by predicting with our mathematical model the effects of a gene inhibition experiment. This thesis ends with the perspective of oriented modulation, *i.e.* targeted interventional experiments on genes allowing to “reprogram” the cancerous genetic program toward a healthy normal state.

Gene regulatory network – Lasso Regression – Systems biology