

THÈSE

présentée pour obtenir

Le titre de Docteur en Sciences
de l'Université d'Évry-Val-d'Essonne
Spécialité : **Mathématiques appliquées**

par

Sophie Lèbre

*Analyse de processus stochastiques pour la génomique :
étude du modèle MTD
et inférence de réseaux bayésiens dynamiques.*

Soutenue le 14 septembre 2007 devant la commission d'examen composée de

<i>Directeur de thèse :</i>	Bernard Prum	Professeur de Statistique à l'Université d'Évry-Val-d'Essonne.
<i>Rapporteurs :</i>	Sylvie Huet	Directeur de Recherche INRA, Unité de Biométrie, Jouy-en-Josas,
	Korbinian Strimmer	Associate Professor for Medical Statistics and Bioinformatics, University of Leipzig.
<i>Examineurs :</i>	Hidde De Jong	Directeur de Recherche INRIA, Unité de Recherche Rhône-Alpes, Grenoble,
	Jacques Istas	Professeur à l'Université Pierre Mendès-France, Département IMSS, Grenoble,
	Catherine Matias	Chargée de Recherche CNRS, Laboratoire Statistique et Génome, Évry-Val-d'Essonne.

Remerciements

Je tiens tout particulièrement à remercier Bernard Prum qui a dirigé ce travail. Bernard, je te remercie d'abord de m'avoir accueillie au sein de ton laboratoire, mais aussi et surtout du soutien et de la confiance que tu m'as accordés au cours de ces trois années. Merci de t'être montré si disponible et réceptif, d'avoir pris le temps de considérer mes multiples questions. Merci pour toutes ces discussions qui m'ont permis de repartir du bon pied à de nombreuses reprises !

Je tiens à remercier Sylvie Huet et Korbinian Strimmer d'avoir accepté de rapporter ma thèse. Je vous suis extrêmement reconnaissante de l'intérêt que vous avez porté à ce travail. Je remercie également Hidde De Jong et Jacques Istas d'avoir accepté de faire partie de mon jury de thèse.

Je remercie tout spécialement Catherine Matias qui m'a consacré beaucoup de temps. Merci Catherine d'avoir été si à l'écoute, de m'avoir soutenue et encouragée tout au long de cette thèse. Le regard rigoureux et critique que tu as porté sur mes travaux a été extrêmement stimulant et enrichissant !

Je souhaite vivement remercier Gaëlle Lelandais pour cette collaboration si enrichissante. C'est un réel plaisir de travailler avec toi et je souhaite que cela continue !

Merci à Pierre-Yves Bourguignon avec qui j'ai réalisé mes premiers travaux de recherche.

Merci à Stéphane Robin d'avoir tant contribué à la communication de mes travaux et à mon ouverture vers l'extérieur.

Merci à Pierre Nicolas pour ses conseils et suggestions pour l'utilisation des MCMC.

Merci à vous les M&Ms, alias Maurice & Mark, de votre aide précieuse pour domestiquer ces machines encore un peu sauvages parfois !

Vincent, je vais faire simple, merci !

Merci à toi Mickaël pour ta jovialité et aussi ton parfait timing. Franck, merci de t'être montré si disponible et clairvoyant, et de toujours communiquer les bonnes infos au bon moment. Merci à toi Carène pour tes conseils avisés et rassurants d'"ex-thésarde". Et plus généralement, je voudrais dire un grand merci à tous les membres du laboratoire Statistique et Génome pour leur soutien, leur bonne humeur et leur disponibilité.

Enfin un grand merci à tous mes proches, bien sollicités eux aussi! Vous m'avez tous montré une confiance énorme et m'avez terriblement encouragée tout au long de cette thèse! Merci à mes parents pour qui c'est une longue histoire et qui me soutiennent depuis un petit moment déjà. Merci à toi Pierrot pour ton soutien téléphonique et musical. Merci Chafika, c'est une chance d'avoir vécu cette expérience en même temps que toi, merci pour le coaching et les soirées parenthèses dans la vie de deux thésardes. Merci à toi Delphine pour tous ces moments RER. Merci Stéphanie d'avoir toujours été là dans les moments importants, et notamment au début de cette thèse. Merci Julie pour beaucoup de choses! Enfin merci à toi, mon cher Matthieu, pour avoir fait que cette thèse se passe avec presque, si j'ose dire, tant de légèreté.

Stochastic process analysis for Genomics and Dynamic Bayesian Networks inference.

Abstract

This thesis is dedicated to the development of statistical and computational methods for the analysis of DNA sequences and gene expression time series.

First we study a parsimonious Markov model called *Mixture Transition Distribution (MTD)* model which is a mixture of Markovian transitions. The overly high number of constraints on the parameters of this model hampers the formulation of an analytical expression of the Maximum Likelihood Estimate (MLE). We propose to approach the MLE thanks to an EM algorithm. After comparing the performance of this algorithm to results from the literature, we use it to evaluate the relevance of MTD modeling for bacteria DNA coding sequences in comparison with standard Markovian modeling.

Then we propose two different approaches for genetic regulation network recovering. We model those genetic networks with Dynamic Bayesian Networks (DBNs) whose edges describe the dependency relationships between time-delayed genes expression. The aim is to estimate the topology of this graph despite the overly low number of repeated measurements compared with the number of observed genes.

To face this problem of dimension, we first assume that the dependency relationships are homogeneous, that is the graph topology is constant across time. Then we propose to approximate this graph by considering partial order dependencies. The concept of partial order dependence graphs, already introduced for static and non directed graphs, is adapted and characterized for DBNs using the theory of graphical models. From these results, we develop a deterministic procedure for DBNs inference.

Finally, we relax the homogeneity assumption by considering the succession of several homogeneous phases. We consider a multiple changepoint regression model. Each changepoint indicates a change in the regression model parameters, which corresponds to the way an expression level depends on the others. Using reversible jump MCMC methods, we develop a stochastic algorithm which allows to *simultaneously* infer the changepoints location and the structure of the network within the phases delimited by the changepoints.

Validation of those two approaches is carried out on both simulated and real data analysis.

Keywords: Time series, Gene expression, Genetic networks, Network inference, Dynamic Bayesian Networks, DBN, Changepoints detection, Reversible jump MCMC, Partial order dependence, Mixture Transition Distribution, MTD, EM algorithm.

Table des matières

1	Introduction	15
1.1	Objectif : comprendre le rôle de chaque gène au sein d'une cellule	15
1.1.1	Spécificité de l'expression d'un gène	15
1.1.2	Un système organisé en réseaux de régulation.	16
1.2	Motivations : données temporelles d'expression de gènes.	17
1.2.1	Puces à ADN ou microarrays	17
1.2.2	Protéines de fluorescence verte (GFP)	18
1.2.3	Délétion de gène ou <i>knockout</i>	19
1.2.4	Normalisation et pré-traitement	19
1.3	Hypothèses et enjeux	21
1.3.1	ARN ou protéine, témoin de l'état fonctionnel d'un organisme.	21
1.3.2	Modéliser un phénomène temporel	21
1.3.3	Faire face à la dimension du problème	21
1.4	Une étape préliminaire : détecter les gènes au sein d'un génome	22
1.4.1	Modélisation markovienne de séquences	22
1.4.2	Des modèles HMM pour représenter l'hétérogénéité des séquences	22
1.4.3	Modélisation <i>parcimonieuse</i> de séquences homogènes	23
1.5	Approches statistiques pour la reconstruction de réseaux génétiques	25
1.5.1	Modélisation dynamique des motifs de régulation	25
1.5.2	Considérer des indépendances partielles pour inférer un réseau bayésien dynamique de grande dimension	27
1.5.3	Reconstruire un réseau <i>chronologique</i> par MCMC à sauts réversibles	28
2	An EM algorithm for estimation in the Mixture Transition Distribution model	31
2.1	Introduction	31
2.2	Upper bound of the MTD model dimension	35
2.3	EM Estimation	37
2.3.1	Introduction of a hidden process	37
2.3.2	EM algorithm	38
2.4	Real data analysis	41
2.4.1	Comparison with Berchtold's Estimation	41
2.4.2	Estimation on DNA coding sequences	43
2.5	Appendix	44
2.5.1	Example of equivalent parameters defining the same MTD ₁ model	44
2.5.2	EM algorithm for other MTD models	45
2.5.3	2 nd order MTD ₁ estimates obtained on both the song of wood pewee and the mouse α A-Crystallin Gene sequence (Section 2.4.1).	46

3	Inferring dynamic genetic networks with low order independencies	49
3.1	Introduction	49
3.2	A DBN representation	52
3.2.1	Backgrounds	53
3.2.2	Sufficient conditions for a DBN representation	54
3.2.3	Minimal DAG $\tilde{\mathcal{G}}$	55
3.2.4	DAG $\tilde{\mathcal{G}}_{AR(1)}$ for an AR(1) process	56
3.3	Approximating $\tilde{\mathcal{G}}$ with DAGs $\mathcal{G}^{(q)}$	57
3.3.1	Definition	57
3.3.2	A restricted number of parents	58
3.3.3	Faithfulness	59
3.4	Inferring $\tilde{\mathcal{G}}$	60
3.4.1	Step 1: inferring $\mathcal{G}^{(1)}$	60
3.4.2	Step 2: inferring $\tilde{\mathcal{G}}$	61
3.5	Validation	63
3.5.1	Simulation study	63
3.5.2	Analysis of microarray time course data sets	65
3.6	Appendix: some additional proofs.	69
4	Inferring time-dependent networks from Systems Biology Time Course Data with reversible jump MCMC.	73
4.1	Introduction	73
4.1.1	Background	73
4.1.2	Non homogeneous network Model	74
4.2	Two Steps Reversible Jump MCMC inference	75
4.2.1	Prior Distributions	77
4.2.2	Parameter space posterior distribution: integration of the “nuisance” parameters (a, σ)	78
4.2.3	Four moves	79
4.3	Simulation study	83
4.4	Analysis of a Microarray Time Course Data Set	85
4.4.1	Early genomic response following benomyl addition data	85
4.4.2	First analysis	86
4.4.3	Conclusion and future work	90
4.5	Appendix	92
4.5.1	Nomenclature	92
4.5.2	Birth of a new predictor for phase h of gene i acceptance probability	92
4.5.3	FLR1: YAP1 early target.	94
4.5.4	GTT2: YAP1 latter target.	96
4.5.5	TPO1: PDR1 target.	98
4.5.6	SNG1: YRR1 and YAP1 target.	100
4.5.7	Estimated posterior distributions for the error variance of GTT2 and SNG1.	102
A	Additional Files: R package ‘G1DBN’ reference manual	103
	inferG1	104
	GfromG1	105
	edges	107
	simulAR1	108

arth800line 109

Chapitre 1

Introduction

1.1 Objectif : comprendre le rôle de chaque gène au sein d'une cellule

1.1.1 Spécificité de l'expression d'un gène

La très grande majorité des fonctions du vivant telles que la respiration, l'alimentation, l'élimination des déchets, la reproduction, ... est assurée par une vaste classe de molécules : les protéines. Celles-ci sont constituées d'une succession de molécules élémentaires qui sont des acides aminés. C'est cette suite d'acides aminés composant chaque protéine qui est "codée" dans le génome de l'organisme, lui-même constitué d'ADN (Acide DésoxyriboNucléique). Une molécule d'ADN se compose de deux longues chaînes en double hélice constituée de quatre *nucléotides* ou bases azotées : adénine, guanine, cytosine et thymine. Seul un très faible pourcentage de l'ADN "code" pour une protéine, moins de 10 % chez l'homme. Nous désignerons ici par "gène" toute partie codante d'un génome.

On estime aujourd'hui que le génome d'un organisme procaryote (organisme sans noyau) tel qu'une bactérie comporte quelques milliers de gènes, tandis que le génome d'un organisme à noyau (un eucaryote) comporte entre environ 6 000 gènes pour la levure *Saccharomyces cerevisiae* jusqu'à environ 30 000 pour l'Homme. Une question essentielle en génomique aujourd'hui est celle de la compréhension du rôle de chaque gène (ou de la protéine associée). Il se trouve que l'organisme ne produit une protéine qu'à la mesure de ses nécessités. En biologie, on parle de régulation de l'expression des gènes. Pour une fonction biologique donnée, seul un "petit" nombre de gènes est utilisé. Dans ce cas, un gène est tout d'abord transcrit, c'est-à-dire que la cellule produit des "copies" simple brin de ce gène : les *ARN* (Acide RiboNucléique). On dit alors que ce gène "s'exprime" : l'ensemble des gènes exprimés change d'un tissu à un autre (du foie aux muscles ou au cerveau), selon les conditions expérimentales (pour une bactérie, selon le pH du milieu, selon les nutriments disponibles, glucose, maltose, saccharose, ...) ou encore selon que l'on ait affaire à une cellule saine ou à une cellule cancéreuse. La plupart des ARN sont des *ARN messagers* (ARNm). Une fois synthétisés, ces ARN sont *traduits* en protéine (Figure 1.1) selon le code génétique : chaque suite de 3 nucléotides, ou *codon*,

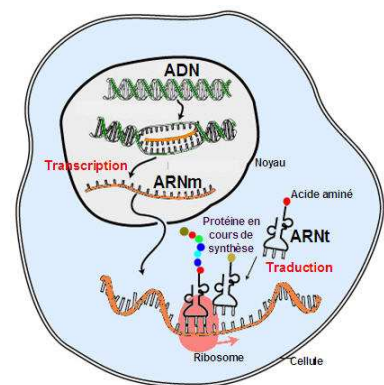


FIG. 1.1 – Du gène à la protéine : transcription et traduction.

code pour un acide aminé spécifique. Un ARNm définit ainsi une suite d'acides aminés qui sont assemblés pour former la protéine correspondante.

1.1.2 Un système organisé en réseaux de régulation.

Malgré la multiplicité des conditions d'expression d'un gène donné, certains groupes de gènes présentent des niveaux d'expression proches dans de nombreuses conditions biologiques. En effet, les protéines n'assurent pas leur rôle dans la cellule de manière individuelle : il peut s'agir de la formation, à un moment donné, de complexes protéiques c'est-à-dire d'amas de plusieurs protéines liées chimiquement. Plus fréquemment, diverses protéines interviennent de façon consécutive dans une cascade de réactions chimiques, on parle alors de *voies métaboliques*. Pour que cette collaboration entre protéines et/ou ARN puisse avoir lieu, il est nécessaire que l'expression des gènes d'un organisme soit contrôlée et coordonnée.

Il existe en effet des mécanismes de régulation de l'expression des gènes. Certaines protéines sont connues pour être des *facteurs de transcription*. Ces protéines favorisent ou au contraire inhibent l'expression d'autres gènes. Elles peuvent se lier à des sites spécifiques, appelés *sites de fixation*, dans les régions régulatrices des gènes. Ces régions peuvent se trouver en amont du gène comme sur la Figure 1.2 mais aussi en aval, et pas nécessairement à proximité du gène. En se fixant à l'ADN, seul ou en formant des complexes les uns avec les autres, les facteurs de transcription peuvent *activer* ou *inhiber* la transcription de leur(s) gène(s) cible(s). Les facteurs de transcription peuvent de plus être eux-mêmes régulés, dans ce cas ils participent à une *voie de régulation*. Ces protéines jouent un rôle clé dans les mécanismes de régulation d'un organisme et la connaissance de certaines d'entre elles représente une information particulièrement intéressante pour la reconstruction d'un réseau de régulation.

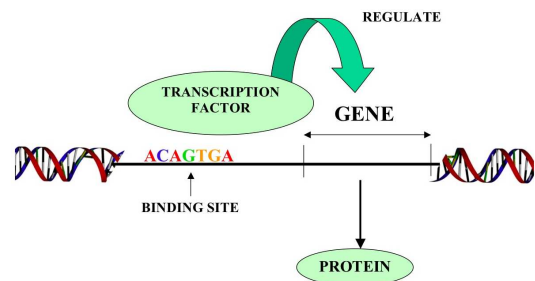


FIG. 1.2 – Activation de la transcription d'un gène.

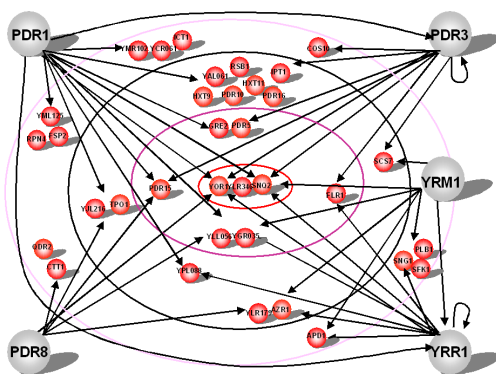


FIG. 1.3 – Réseau PDR de *S. cerevisiae*.

produits toxiques chez la levure *S. cerevisiae*.

Un tel réseau de régulation peut être reconstruit grâce à la mise en oeuvre de différentes techniques comprenant entre autres la détection de gènes, l'identification de facteurs de transcription, la recherche de sites de fixation dans la séquence d'ADN, la détection de la capacité pour deux

Ainsi, un gène cible peut être régulé par une combinaison de facteurs de transcription et un facteur de transcription peut réguler plusieurs gènes cibles. Un des principaux défis actuel est de comprendre ces phénomènes de régulation entre les gènes. Il s'agit de représenter l'ensemble des relations de régulation caractéristiques d'un processus sous la forme d'un graphe dont les noeuds sont les gènes (facteurs de transcription et gènes cibles) et les arêtes orientées représentent des effets transcriptionnels activateurs ou inhibiteurs [WLL04, GF05]. Le réseau "PDR" représenté Figure 1.3 a ainsi été identifié par l'équipe Génomique de la Levure (ENS, Paris) pour la résistance aux

protéines à former un complexe protéique. De plus, nous bénéficions aujourd'hui d'un large panel de données qui nous donne la possibilité d'observer le comportement des gènes. Il s'agit maintenant de développer des méthodes statistiques qui permettent de déterminer ou plus exactement aident à déterminer l'organisation sous-jacente à un processus cellulaire à partir de ces données. Les principaux types de données d'expression disponibles actuellement sont présentés dans la section suivante.

1.2 Motivations : données temporelles d'expression de gènes.

1.2.1 Puces à ADN ou microarrays

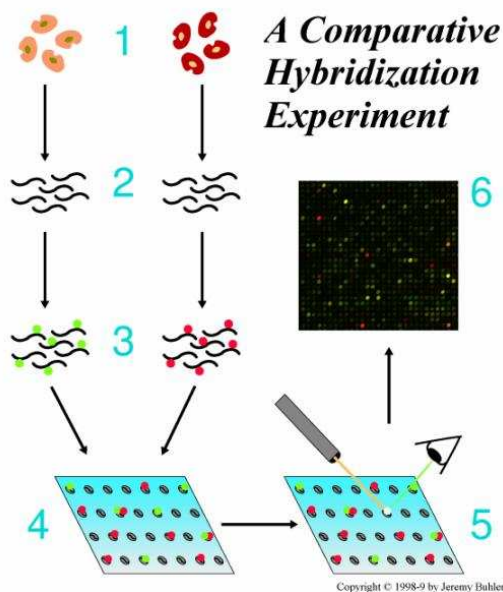


FIG. 1.4 – Synthèse d'une puce à ADN.

Unes méthode extrêmement répandue aujourd'hui consiste à mesurer le niveau d'expression des gènes au moyen de puces à ADN ou *microarrays* [SSDB95] dont une illustration apparaît en Figure 1.5. Une puce à ADN est un ensemble de molécules d'ADN fixées sur une surface qui peut être du verre, du silicium ou bien encore du plastique. Cette biotechnologie récente permet de visualiser les gènes exprimés dans une cellule d'un tissu donné (foie, intestin...), à un moment donné (embryon, adulte...) et dans un état donné (malade, saine...) .

Concrètement, les ARN sont d'abord extrait des cellules. puis transformés en ADN complémentaires (cDNA) par la technique de *reverse transcription*. Ils sont ensuite transformés en ARN complémentaires (cRNA) et marqués par un colorant : la Cyanine 3 (qui fluoresce en vert) ou la Cyanine 5 (en rouge). Une fois marqués, ces ARN complémentaires sont déposés sur la lame de verre qui elle-même possède, fixés à sa surface, des fragments de génome recouvrant un ensemble de gènes présents dans une cellule. Des milliers de molécules d'ADN ou sondes peuvent être fixées sur une même puce. Actuellement, les puces les plus performantes peuvent accueillir jusqu'à 4×44 kilobases d'ADN soit l'équivalent de 4 génomes humains complets. La comparaison de deux expériences (A et B) de puce à ADN sur deux extraits de cellules du même type, l'une saine et l'autre malade par exemple, peut permettre de découvrir des gènes exprimés uniquement dans la cellule saine (ou uniquement dans la cellule malade).

Ainsi, une coloration en fausse couleur témoin la fluorescence observée permet comme sur la figure 1.6 de représenter,

- en rouge les lieux de dépôt ou *spots* associés à un gène s'exprimant davantage dans la condition A que dans la condition B ;
- en vert les spots associés à un gène s'exprimant davantage dans la condition B que dans la condition A ;
- en jaune les spots associés à un gène s'exprimant de façon analogue dans les deux conditions ;
- en noir les spots associés à un gène ne s'exprimant ni dans l'une, ni dans l'autre condition.

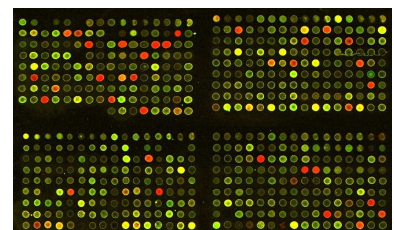


FIG. 1.5 – Puce à ADN.

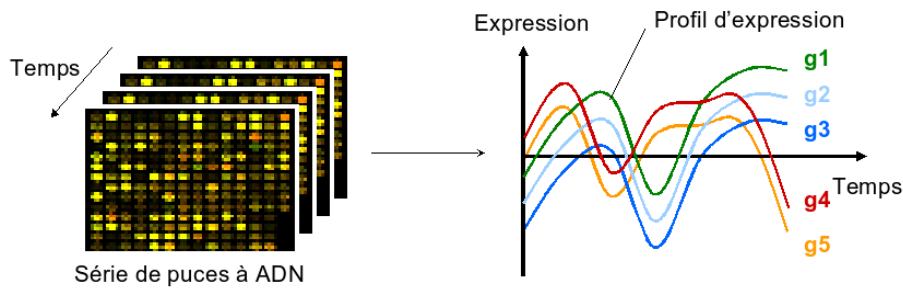


FIG. 1.6 – Profils d’expression temporels obtenus par puces à ADN.

Il existe d’autres types de puce, notamment les puces *Affymetrix* qui utilisent un simple marquage (une seule couleur) et donnent ainsi une mesure absolue de la quantité d’ARNm. Dans tous les cas, les puces à ADN permettent de mesurer le niveau d’expression de milliers de gènes simultanément. De plus, la mesure peut être effectuée à un instant précis. En répétant l’opération sur plusieurs points de temps successifs, on est alors capable de suivre un processus au cours de son évolution. On obtient ainsi un profil d’expression pour chaque gène comme illustré en Figure 1.6.

Par exemple, on synchronise le cycle cellulaire d’un très grand nombre de cellules de levure *Saccharomyces cerevisiae* et l’on mesure l’expression des gènes chez ces cellules alors qu’elles se trouvent toutes à un instant précis du cycle de reproduction. Un des premiers jeux de données ainsi obtenu est celui de Spellman et al. [SSZ⁺98]. Le niveau d’expression de 6 275 gènes est mesuré sur 18 points de temps régulièrement espacés (7 min). On observe ainsi environ deux cycles cellulaires.

Afin d’étudier le métabolisme de l’amidon chez la plante *Arabidopsis thaliana*, Smith et al. [SFC⁺04] ont mesuré le niveau d’expression de ses 22 810 gènes sur un cycle de 24h (11 points de temps, 2 répétitions). Les différentes approches pour la reconstruction de réseaux de régulation proposées dans cette thèse seront notamment appliquées à ces deux jeux de données.

1.2.2 Protéines de fluorescence verte (GFP)

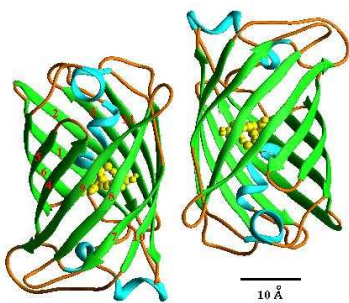


FIG. 1.7 – Protéines de fluorescence verte (GFP).

La protéine de fluorescence verte (GFP pour *Green Fluorescent Protein*) est intrinsèquement fluorescente. Le procédé consiste à fusionner le gène de cette protéine au gène d’une protéine que l’on souhaite observer. On peut alors mesurer la quantité de cette protéine produite dans la cellule à l’aide d’un microscope à fluorescence.

Un avantage majeur de cette technique est de permettre l’étude des protéines dans leur environnement naturel : la cellule reste vivante. Il est de plus possible d’observer la fluorescence à des temps très rapprochés. [RRS02] ont mesuré le taux de synthèse de huit protéines impliquées dans le système de réparation de l’ADN de *E. Coli*. Les mesures de fluorescence ont été effectuées sur 50 points de temps équirépartis à 6 min d’intervalle.

Cependant, il faut noter que cette technique nécessite une intervention spécifique (fusion du gène de la GFP) pour chaque protéine étudiée. Il est de plus nécessaire de connaître le taux de dégradation de la GFP pour obtenir le taux de synthèse propre à la protéine étudiée.

1.2.3 Délétion de gène ou *knockout*

On est depuis longtemps capable “d’éteindre” un gène, c’est-à-dire d’empêcher son expression. C’est ce que l’on nomme délétion ou *knockout*. Grâce à cette technique, il est possible de comparer par puces à ADN l’expression de l’ensemble des gènes d’un organisme selon que l’un d’entre eux est délété ou non. L’avantage de ce type de données est de permettre la mise en évidence de phénomènes de régulation non-transcriptionnelle, c’est-à-dire de modes de régulation qui ne sont pas dus à la quantité de la protéine facteur de transcription présente dans la cellule. Dans ce cas il est impossible de déceler l’implication de cette protéine dans un processus de régulation en mesurant la quantité d’ARNm correspondant. La seule façon de comprendre l’effet d’une telle protéine est d’étudier des cellules qui sont incapables de la synthétiser.

L’approche pour la reconstruction du réseau de régulation de la Levure *S. cerevisiae* après ajout d’une drogue (le bénomyl) développée section 4.4 est obtenue à partir de données knockout. En effet, il a été mis en évidence notamment que l’un de facteurs de transcription impliqués dans le processus de réaction au bénomyl, la protéine YAP1, régule l’expression de ses gènes cibles en fonction de sa localisation au sein de la cellule (noyau ou cytoplasme).

1.2.4 Normalisation et pré-traitement

La réalisation de puces à ADN nécessite un certain nombre d’étapes. Le biologiste doit notamment déterminer quels gènes ou séquences codantes déposer sur les puces, quelles cellules utiliser comme source d’ARN à hybrider sur ces puces et combien de répétitions effectuer.

Pour les puces les plus utilisées, celles en “rouge et vert”, on mesure ensuite pour chaque spot l’intensité de signal rouge, de signal vert ainsi que le bruit de fond. Pour cela, l’expérimentateur doit choisir le réglage du laser. En particulier la forme de la zone sur laquelle l’intensité est mesurée joue un rôle déterminant. La méthode de segmentation la plus simple consiste à mesurer l’intensité de fluorescence sur un cercle de diamètre constant. Or les spots sont rarement parfaitement circulaires et de même taille. Aussi existe-t-il d’autres méthodes qui permettent d’estimer le diamètre de chaque spot. Cela est notamment proposé par les deux logiciels *GenePix* et *Dapple*. Il existe aussi des méthodes de segmentation qui ne reposent pas sur une segmentation circulaire [BM93, AB94].

Après cela, il s’en suit une série de premières étapes de “nettoyage” des données. On peut en effet choisir de corriger l’intensité du signal mesuré en fonction du bruit de fond. Puis, le ratio Rouge/Vert ainsi obtenu peut être normalisé afin de s’affranchir des différents biais survenus au cours de l’expérience (effet puce, effet bloc, ...). Un biais caractéristique des puces à ADN est le biais rouge/vert. En effet, la fluorescence rouge est globalement supérieure à la fluorescence verte. Ce biais est généralement corrigé au moyen d’une régression LOESS ou LOWESS (*Local weighted regression*) [YDLS01, YBDS02]. Pour cela, on s’appuie sur l’hypothèse que peu de gènes s’expriment différemment entre les deux conditions. Dans ce cas, la quantité de fluorochromes incorporée n’a pas d’influence sur le rapport $\log(R/V)$. Par conséquent, sur le graphe M-A, qui représente la différence des signaux ($M = \log(R) - \log(V)$) en fonction du signal moyen mesuré pour chaque spot $A = (\log(R) + \log(V))/2$ où R est le signal en rouge et V le signal en vert, le nuage de points devrait se situer autour de l’axe des abscisses. Cela permet de corriger les log-ratios de chaque spot après avoir estimé la courbe c de tendance $\log(R) - \log(V) = c\left(\frac{\log(R) + \log(V)}{2}\right)$. L’intérêt de cette correction est illustré figure 1.8.

Il s’agit ensuite de déterminer, parmi le très grand nombre de gènes observés, quels sont les gènes différemment exprimés au cours de l’expérience. Pour cela, un grand nombre de méthodes ont été utilisées, à commencer par la sélection des gènes dont le log-ratio est supérieur à

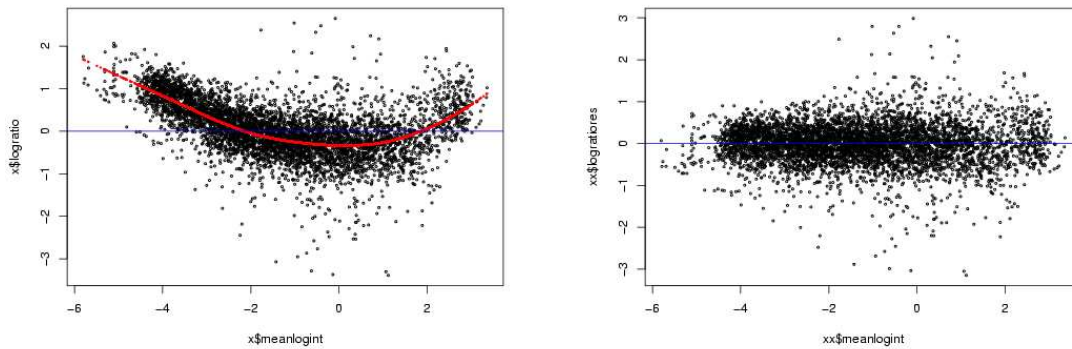


FIG. 1.8 – Graphe M-A : avant et après correction du biais rouge/vert.

deux écart-types, ou bien au moins deux fois supérieur à deux écart-types au cours de la cinétique... on pourrait ainsi définir arbitrairement autant de seuils. Une approche générale consiste à ordonner les gènes selon une statistique mettant en évidence leur caractère différentiel d'expression puis à définir un seuil au-dessus duquel les gènes sont considérés différentiellement exprimés de manière significative. Un choix plus judicieux, et aussi plus répandu, consiste à ordonner les gènes selon la statistique de Student,

$$t = \frac{\overline{M}}{s/\sqrt{n}},$$

où \overline{M} est la moyenne des log-ratios obtenus sur les n réplicats d'un gène et s l'écart-type. Dans ce cas, il s'agit d'estimer correctement la variance pour chaque gène. Or le nombre de réplicats est souvent trop faible pour permettre une bonne estimation. Une alternative consiste à utiliser un modèle de mélange, c'est-à-dire à réaliser une partition de l'ensemble des gènes ayant la même variance [DRD05]. Cela permet un compromis efficace entre l'hypothèse d'homoscédasticité trop réductrice et le modèle surparamétré considérant une variance propre à chaque gène. Quelle que soit la méthode choisie, il s'agit ensuite de prendre en compte les tests multiples pour fixer le seuil de sélection des gènes différentiellement exprimés.

Par ailleurs, si l'on observe un processus cyclique, on peut chercher à détecter les gènes dont le niveau d'expression est lui aussi cyclique. Wichert et al. [WFS04] proposent pour cela un test de périodicité. C'est notamment selon cette méthode que les 800 gènes, sur lesquels porte l'application développée dans la section 3.5.2, ont été sélectionnés parmi les données sur le métabolisme diurne de l'amidon chez *Arabidopsis thaliana* [SFC⁺04].

Ainsi, la normalisation des données tout comme la sélection des gènes différentiellement exprimés constitue un objet d'étude en soi. Aussi le déroulement de ces deux étapes préliminaires ne sera-t-il pas davantage approfondi ici. Pour en savoir plus, Smyth et al. [SYS03] proposent une étude complète des questions de normalisation de données de puces à ADN ainsi que de nombreuses références. Les différentes analyses de données réelles présentées dans cette thèse sont effectuées sur des données publiées. La normalisation des données puis la sélection des gènes ont alors été effectuées préalablement. Un des principaux enjeux de cette thèse démarre précisément à l'issue de ces deux étapes : comment analyser ces données une fois que l'on s'est affranchi - au mieux - des différentes sources de bruit, qu'elles soient d'origine biologique ou technologique ?

1.3 Hypothèses et enjeux

1.3.1 ARN ou protéine, témoin de l'état fonctionnel d'un organisme.

La mesure à grande échelle de l'expression génique est motivée notamment par l'hypothèse que l'état fonctionnel d'un organisme est en grande partie décrit par la quantité de chaque ARN présent dans la cellule à un instant donné. Dans le cas d'une régulation transcriptionnelle, la quantité d'ARN d'un gène cible évolue en fonction de la quantité d'ARN de son (ses) facteur(s) de transcription. Il est donc possible d'étudier les relations entre gènes cibles et facteurs de transcription à partir de la mesure par puces à ADN de leurs niveaux d'expression [SK99, BdIFM02].

L'analyse peut aussi être réalisée à partir de données obtenues par GFP qui reflètent le taux de synthèse d'une protéine au sein d'un organisme vivant (voir section 1.2.2). Il est beaucoup plus long et coûteux de produire des données GFP pour un grand nombre de protéines mais cette technique se développe et commence à être automatisée. On est donc en mesure de pouvoir rapidement obtenir des données GFP à grande échelle.

En revanche, si le mode de régulation d'un facteur de transcription n'est pas transcriptionnel, il est alors nécessaire "d'éteindre" le gène correspondant (voir section 1.2.3) pour observer une modification du niveau d'expression des gènes cibles. L'effet de chaque délétion peut alors être quantifié au moyen de puces à ADN.

1.3.2 Modéliser un phénomène temporel

D'une manière générale, on est donc amené à considérer "l'activité" simultanée d'un grand nombre de gènes ou de protéines, et ce, tout au long du processus étudié. Il s'agit alors d'exploiter cette information pour proposer des éléments d'aide à la détection des phénomènes de régulation responsables de ce processus. Il serait en effet bien ambitieux de souhaiter reconstruire exactement *le* réseau de régulation sous-jacent. On peut néanmoins effectuer une pré-sélection parmi l'ensemble des interactions possibles et ainsi mettre en évidence les interactions "les plus probables" au vu des données observées. Cela permet alors au biologiste de restreindre le nombre d'interactions potentielles à étudier, et ainsi de réaliser une analyse approfondie des gènes et interactions mis en évidence par une approche statistique.

Pour cela, il est tout d'abord nécessaire de proposer une modélisation capable de saisir des relations de dépendances entre les différents gènes observés. En effet, la significativité des résultats dépend de la modélisation choisie. Aussi le choix ou la construction du modèle doit-il répondre à deux exigences essentielles : offrir une bonne qualité d'estimation tout en saisissant au mieux les caractéristiques des phénomènes biologiques de régulation. Dès lors il s'agit de réaliser un compromis entre le niveau de description du modèle et sa robustesse.

Pour bien décrire les phénomènes de régulation responsables du processus observé, l'information temporelle doit jouer un rôle essentiel dans la modélisation. Cela commence par la prise en compte la dépendance existant entre deux mesures successives.

1.3.3 Faire face à la dimension du problème

Un enjeu essentiel est de faire face à la dimension du problème : grâce aux puces à ADN, on observe le niveau d'expression de milliers de gènes simultanément. Même après une pré-sélection des gènes différentiellement exprimés ou dont le niveau d'expression est cyclique (voir section 1.2.4), le nombre de gènes p à étudier reste de plusieurs centaines à quelques milliers (786 gènes pour le cycle cellulaire de la Levure, 119 gènes pour la réaction au bénomyl, 800 gènes pour l'étude

du métabolisme de l'amidon chez *Arabidopsis Thaliana*). Sans connaissance a priori des facteurs de transcription, le nombre d'interactions potentielles est de p^2 alors que l'on ne dispose que de np mesures d'expression (n étant le nombre de répétitions pour chaque gènes) et que pour l'instant n est très petit (< 20 en général).

Afin de tirer le meilleur parti de ces données, il s'agit donc de proposer une modélisation et surtout une méthode d'estimation qui permettent à la fois de considérer un grand nombre de variables et d'obtenir des résultats significatifs en dépit de ce déséquilibre entre la quantité de données disponibles et la dimension du modèle.

1.4 Une étape préliminaire : détecter les gènes au sein d'un génome

1.4.1 Modélisation markovienne de séquences

Une séquence d'ADN est constituée d'une succession orientée de nucléotides (adénine, guanine, cytosine et thymine). Une séquence de longueur n sera ainsi représentée par la suite de variables aléatoires $\mathbf{Y} = (Y_1, \dots, Y_n)$ prenant ses valeurs dans un alphabet discret $\mathcal{Y} = \{1, \dots, q\}$ où $q = 4$. Un premier objectif consiste à déterminer quelles sont les régions codantes au sein de cette séquence. Ces régions ont des propriétés de composition différentes des régions non-codantes ou *intergéniques* [Nic03]. La plupart des méthodes intrinsèques pour mettre en évidence les régions codantes utilisent cette propriété.

Si l'on considère que la distribution des nucléotides est homogène tout au long de la séquence d'ADN, le modèle le plus simple consiste à supposer que les nucléotides Y_t sont générés de manière identique et indépendante. La modélisation par *chaîne de Markov* d'ordre 1 - ou plus généralement d'ordre m - est plus raisonnable car elle intègre les dépendances locales qui peuvent exister entre les nucléotides. Notons,

$$\mathbf{Y}_{t_1}^{t_2} = (Y_{t_1}, Y_{t_1+1}, \dots, Y_{t_2}),$$

la sous-suite de $t_2 - t_1 + 1$ variables successives. Une séquence aléatoire \mathbf{Y} est une chaîne de Markov d'ordre m si,

$$\begin{aligned} \forall t > m, \forall y_1, \dots, y_t \in \mathcal{Y}, \quad \mathbb{P}(Y_t = y_t | \mathbf{Y}_1^{t-1} = \mathbf{z}_1^{t-1}) &= \mathbb{P}(Y_t = y_t | Y_{t-m}^{t-1} = y_{t-m}^{t-1}), \\ &= \pi(y_{t-m}, \dots, y_{t-1}; y_t), \end{aligned}$$

où π est une matrice stochastique de taille $q^m \times q$, c'est-à-dire, que cette matrice satisfait,

$$\forall y_{t-m}, \dots, y_{t-1} \in \mathcal{Y}, \quad \sum_{y_t=1}^{y_t=q} \pi(y_{t-m}, \dots, y_{t-1}; y_t) = 1 \quad \text{et} \quad \pi(y_{t-m}, \dots, y_{t-1}; y_t) \geq 0.$$

1.4.2 Des modèles HMM pour représenter l'hétérogénéité des séquences

L'ajustement d'un même modèle d'un bout à l'autre de la séquence ne saurait refléter l'hétérogénéité qui peut exister entre les régions de natures différentes au sein de la séquence, notamment entre le codant et l'intergénique. Aussi, une approche consiste à proposer pour cette séquence deux modélisations : l'une pour les régions codantes, l'autre pour les régions intergéniques. Dès lors, on peut chercher à distinguer ces deux natures de région selon qu'une région est "mieux" représentée

par l'un ou l'autre des modèles. On généralise au cas de M natures ($M > 2$) pour une description plus précise. Il s'agit alors "d'apprendre" les modèles pour chaque nature de région et d'affecter chaque élément de la séquence à la nature la plus probable.

Les modèles de chaînes de Markov *cachées* (HMM pour *Hidden Markov Model*) permettent d'accomplir ces deux étapes simultanément [Mur97, Nic03]. Cette modélisation repose sur l'hypothèse que la probabilité d'occurrence de chaque lettre est spécifique à chaque type de région. Le modèle HMM introduit une variable cachée S_t qui décrit la nature de la séquence à la position t . Or, la nature d'un élément de la séquence dépend de la nature des éléments adjacents. C'est en tenant compte de ce contexte que l'information sur une position particulière peut être enrichie.

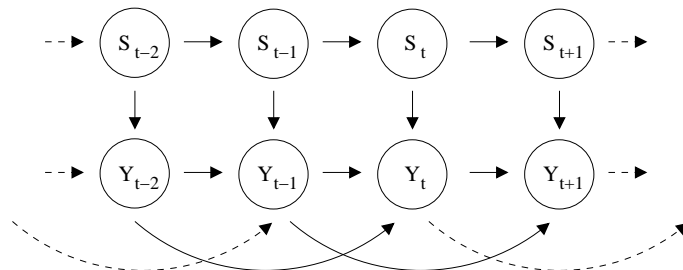


FIG. 1.9 – Structure de dépendance pour un modèle HMM $M1-M2$. Chaque variable S_t est une variable "cachée" qui décrit la nature de la séquence à la position t .

Les HMM modélisent cette dépendance sous la forme d'une chaîne de Markov $(S_t)_{t=1..n}$ sur la nature de chaque position de la séquence. On définit ainsi un modèle simple qui décrit d'une part les probabilités de transition "d'une nature à une autre" et d'autre part, la loi d'apparition $P(Y_t|S_t)$ de chaque lettre Y_t conditionnellement à la nature de la séquence S_t à la position t . Cette structure est représentée Figure 1.9. Les HMM permettent ainsi de modéliser très librement l'alternance de textures et de signaux le long des séquences. L'estimation d'un tel modèle permet de détecter la nature des différentes régions au sein d'une séquence et en particulier de mettre en évidence des régions codantes.

On peut de plus chercher à tenir compte du contexte pour définir les lois d'apparition des variables observées Y_t et ainsi supposer que la loi de Y_t dépend des m variables précédentes Y_{t-m}^{t-1} selon un modèle markovien. Ces modèles HMM du type $M1-Mm$ (1 pour la dépendance markovienne des régimes cachés S_t , m pour celle des variables observées Y_t) ont été introduits par Churchill [Chu89] et développés par Muri [Mur97] pour l'analyse de séquences génomiques. Le schéma de dépendance dans le cas d'un modèle HMM $M1-M2$ apparaît en Figure 1.9.

1.4.3 Modélisation *parcimonieuse* de séquences homogènes

Concentrons nous maintenant sur la modélisation de séquences - ou de portions de séquences - homogènes, c'est-à-dire correspondant à un même régime. Pour extraire un maximum d'information de la séquence, on peut chercher à utiliser un modèle markovien ayant une mémoire m longue. En revanche, lorsque l'ordre du modèle augmente, le nombre de paramètres croît de façon exponentielle et la qualité d'estimation du modèle se voit d'autant détériorée. Il s'agit alors de proposer une modélisation parcimonieuse qui permette de maintenir une bonne qualité d'estimation. Différentes approches ont été proposées pour réduire le nombre de paramètres. On peut citer notamment les modèles de Markov à longueur de mémoire variable (VLMC pour *Variable Length Markov Chains*) [BW99] et les "modèles de Markov parcimonieux" [BR04]. Je propose d'étudier, dans le chapitre 2, l'intérêt du modèle MTD ou *Mixture Transition Distribution* pour la modélisation de séquences homogènes. Le modèle MTD est un modèle markovien, défini par

un mélange de transitions markoviennes. La séquence aléatoire \mathbf{Y} suit un modèle MTD d'ordre m si,

$$\begin{aligned} \forall t > m, \forall y_1, \dots, y_t \in \mathcal{Y}, \quad \mathbb{P}(Y_t = y_t | \mathbf{Y}_1^{t-1} = \mathbf{z}_1^{t-1}) &= \sum_{g=1}^m \varphi_g \mathbb{P}(Y_t = y_t | Y_{t-g} = y_{t-g}) \\ &= \sum_{g=1}^m \varphi_g \boldsymbol{\pi}_g(y_{t-g}, y_t), \end{aligned}$$

où le vecteur $\boldsymbol{\varphi} = (\varphi_1, \dots, \varphi_m)$ satisfait les deux contraintes suivantes :

$$\forall g \in \{1, \dots, m\}, \varphi_g \geq 0 \quad \text{et} \quad \sum_{g=1}^m \varphi_g = 1,$$

et les matrices $\{\boldsymbol{\pi}_g = [\mathbb{P}(Y_t = j | Y_{t-g} = i)]_{i,j \in \mathcal{Y}}; 1 \leq g \leq m\}$ sont des matrices stochastiques de taille $q \times q$. Cette modélisation suppose une contribution additive de chacune des m lettres précédant celle à prédire.

Lorsque la mémoire m est longue, le modèle MTD permet de réduire considérablement le nombre de paramètres. Cependant, le modèle tel qu'il est décrit par les paramètres $(\boldsymbol{\varphi}, \boldsymbol{\pi})$ n'est pas identifiable. En revanche, un modèle MTD décrit un modèle de Markov "complet" qui, lui, est identifiable. Simplement, comme le montre l'exemple cité en section 2.5.1, un même modèle MTD peut être décrit par plusieurs jeux de paramètres $(\boldsymbol{\varphi}, \boldsymbol{\pi})$. En conséquence, pour parler d'une *classe* de modèles MTD d'ordre m , on ne considérera pas les paramètres $(\boldsymbol{\varphi}, \boldsymbol{\pi})$ mais la matrice de transition de taille $q^m \times q$ qui définit le modèle markovien correspondant. On peut noter que la dimension du modèle est encore inférieure au nombre de paramètres libres parmi les éléments $(\boldsymbol{\varphi}, \boldsymbol{\pi})$. En effet, un sous ensemble de ces paramètres suffit à définir un modèle MTD. L'étude de la dimension du modèle MTD fait l'objet de la section 2.2.

Le problème posé par l'utilisation du modèle MTD se trouve dans l'estimation même de ce modèle. En effet, la forme de la vraisemblance ne permet pas de maximisation directe. Différentes approches fondées sur des méthodes d'approximation numérique ont été proposées [LK90, RT94]. Berchtold [Ber01] a introduit une méthode efficace mais qui repose tout d'abord sur l'hypothèse forte d'indépendance des paramètres de poids $\boldsymbol{\varphi}$ et des matrices de transition $\{\boldsymbol{\pi}_g\}$. En effet, la vraisemblance est maximisée selon le vecteur $\boldsymbol{\varphi}$ puis selon chaque matrice $\boldsymbol{\pi}_g$ de manière itérative. De plus, cette procédure nécessite de choisir arbitrairement un certain nombre de paramètres.

J'introduis dans la section 2.3 un algorithme EM (Expectation Maximisation) pour l'inférence des modèles MTD. Cet algorithme est très simple à utiliser, s'adapte à tous les types de MTD (différentes mémoires, différentes composantes de mélanges) et offre des propriétés de convergence. Cet algorithme est notamment utilisé pour la modélisation de séquences codantes de bactéries en section 2.4.2. Si l'on se réfère au classique critère BIC ou *Bayesian Information Criterion* de pénalisation de la vraisemblance, le modèle MTD se révèle particulièrement pertinent et surclasse les modèles de Markov dès que l'ordre du modèle est supérieur à 5.

Le modèle MTD offre ainsi des perspectives intéressantes pour la détection de gènes. Il conviendrait pour cela d'étendre l'algorithme EM développé dans la section 2.3 à l'estimation de modèles HMM dans lesquels les lois d'apparition sont définies par des modèles MTD. Il serait alors possible de prendre en compte un contexte plus long pour détecter les gènes au sein d'une séquence d'ADN.

1.5 Approches statistiques pour la reconstruction de réseaux génétiques

1.5.1 Modélisation dynamique des motifs de régulation

Une fois que l'on a déterminé un ensemble de gènes à étudier, il s'agit de déterminer le rôle de chacun. Les puces à ADN permettent d'observer l'expression de l'ensemble de ces gènes simultanément et offrent ainsi un point de vue global sur le comportement de ce système. Ceci constitue une source de données considérable dont il s'agit de tirer le maximum d'information. Pour cela, on introduit le processus X décrivant l'intensité d'expression de p gènes sur n instants successifs,

$$X = \{X_t^i; 1 \leq i \leq p, 0 \leq t \leq n\},$$

où chaque variable aléatoire X_t^i représente l'intensité d'expression du gène i à l'instant t . Ainsi le vecteur X_t , de dimension p , décrit le niveau d'expression de l'ensemble des gènes observés au temps t et le vecteur X^i , de dimension n , le profil d'expression du gène i tout au long de l'expérience. Il s'agit alors d'appréhender les relations qui existent entre les différents gènes observés à partir de l'observation de ce processus.

Jusqu'ici, un grand nombre d'approches ont été utilisées pour décrire des systèmes de régulation et en particulier dans le but de reconstruire un graphe, orienté ou non-orienté (voir [DJ02, BJ05] pour une synthèse). Il s'agit de mettre en évidence des motifs de régulation comme celui représenté Figure 1.10 par exemple. Une arête tracée du gène G^1 vers le gène G^2 traduit le fait que la protéine codée par le gène 1 (facteur de transcription) régule l'expression du gène 2.

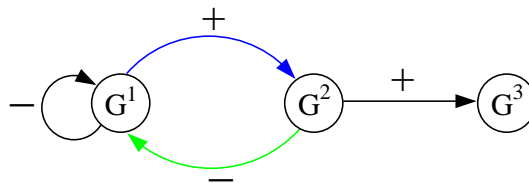


FIG. 1.10 – Exemple de motif de régulation.

Une approche assez communément établie aujourd'hui repose sur les modèles graphiques gaussiens ou GGM pour *Graphical Gaussian Model* [SS05a, SS05b, WK00a, TH02a, TH02b, WYS03, WMH03]. On suppose pour cela que l'ensemble des niveaux d'expression des p gènes observés est un vecteur gaussien $(Y^i)_{i=1..p}$ dont on observe des répétitions. Chaque gène i est représenté par un noeud Y^i et l'on trace une arête entre deux noeuds dès que les niveaux d'expression correspondants Y^i et Y^j sont corrélés conditionnellement à l'ensemble des niveaux d'expression observés. Le graphe ainsi obtenu est appelé *graphe de concentration*. Si l'on observe un phénomène gouverné par le motif de la Figure 1.10 par exemple, on s'attend à inférer le graphe représenté Figure 1.11. Ce graphe permet bien de retrouver que la dépendance entre les variables Y^1 et Y^3 se fait par l'intermédiaire de la variable Y^2 .

L'approche GGM permet en effet de détecter *uniquement* les relations de dépendance *directes* entre les différents niveaux d'expression. On évite ainsi de tracer une arête entre deux gènes dont les niveaux d'expression sont corrélés en raison de leur lien respectif avec un troisième gène par exemple. En revanche, il n'est pas possible de représenter certains motifs comme les boucles de

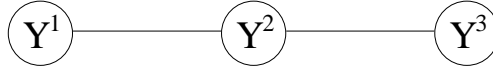


FIG. 1.11 – Graphe de concentration (GGM) correspondant au motif de la Figure 1.10.

rétroaction (auto-inhibition d’un gène, e.g. gène G1 Figure 1.10) ou les boucles “multicomponent” (e.g. motif entre les gènes G1 et G2, Figure 1.10) qui ont été mis en évidence dans les mécanismes de régulation de la levure par exemple [LRR⁺02]. En outre, l’aspect dynamique du système n’est pas pris en compte et l’on ne dispose pas d’indication sur l’orientation des arêtes.

C’est notamment ce qui a suscité un grand intérêt pour la modélisation de *réseaux bayésiens dynamiques* [FMR98, MM99]. Ces modèles probabilistes sont définis par un graphe orienté (Figure 1.12) dont les noeuds sont les variables X_t^i représentant les niveaux d’expression de chaque gène i à chaque instant de mesure t . Pour permettre la définition d’un réseau bayésien, ce graphe doit être *acyclique*. Ainsi, le processus X admet une représentation selon un réseau bayésien défini par un graphe acyclique orienté (ou DAG pour *Directed Acyclic Graph*) dès que la loi jointe de ce processus s’écrit comme le produit des densités conditionnelles de chaque variable sachant ses parents dans ce graphe.

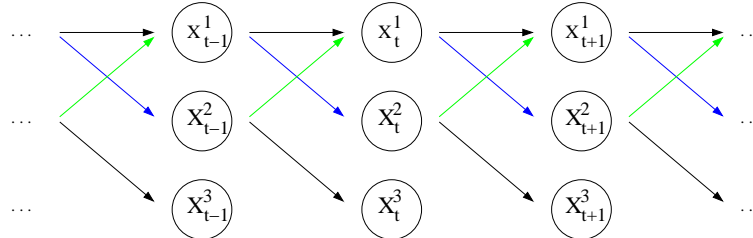


FIG. 1.12 – Exemple de graphe orienté acyclique ou DAG définissant un réseau bayésien dynamique. Chaque noeud X_t^i représente le niveau d’expression du gène i au temps t . Si l’on suppose un délai Δ_t constant, ce graphe est exactement équivalent au motif représenté en Figure 1.10.

Un réseau bayésien défini par un DAG tel que celui de la Figure 1.12 est qualifié de *dynamique* car chaque gène est non pas représenté par un seul noeud mais par autant de noeuds qu’il y a de points de mesure. C’est ce qui permet de prendre en compte la dépendance temporelle entre les variables. De plus, l’immense avantage de cette modélisation est de permettre la représentation de tout motif biologique classiquement défini par un graphe orienté tel que celui de la Figure 1.10. En effet, un réseau bayésien ne peut être défini que par un graphe acyclique. Or, on s’attend à ce qu’un grand nombre de motifs biologiques contiennent des cycles (rétroaction, boucles “multicomponent”). Le graphe de la Figure 1.10 ne peut donc pas être directement utilisé pour définir un réseau bayésien. En revanche, si l’on considère que les relations de régulation s’effectuent avec un certain délai, c’est-à-dire qu’une boucle de rétroaction par exemple se traduit par l’effet du niveau d’expression d’un certain gène au temps $t - 1$ sur son niveau d’expression au temps suivant, le graphe représenté en Figure 1.12 est exactement équivalent à celui de la Figure 1.10.

1.5.2 Considérer des indépendances partielles pour inférer un réseau bayésien dynamique de grande dimension

Introduite par Murphy et al. [MM99], la modélisation de réseaux bayésiens dynamiques pour l’analyse de données d’expression de gènes a depuis suscité un intérêt croissant. Ainsi, un très grand nombre de modèles a été proposé ; notamment un modèle auto-régressif multi-dimensionnel [ORS07], des réseaux booléens [OGP02, ZC05], des chaînes de Markov cachées [PRM⁺03, WZK04, RAG⁺04, BFG⁺05], des modèles additifs de régression non-paramétrique [IGM02, IKG⁺03, KIM04, SI04].

Aussi, je propose dans un premier temps d’établir un cadre général qui permet la définition d’un réseau bayésien dynamique (DBN pour *Dynamic Bayesian Network*), indépendamment de la “forme” choisie pour le modèle. Il s’agit de mettre en évidence les hypothèses sous-jacentes à cette modélisation. J’expose pour cela dans la section 3.2 des conditions suffisantes pour que le processus X admette une représentation selon un DBN défini par un unique DAG $\tilde{\mathcal{G}}$. Ce DAG $\tilde{\mathcal{G}}$ décrit exactement les relations de dépendance conditionnelle entre deux niveaux d’expression successifs sachant le passé du processus. Lorsqu’une arête de ce DAG $\tilde{\mathcal{G}}$ est tracée du noeud X_{t-1}^1 vers X_t^2 , le gène 1 est alors mis en évidence comme un facteur de transcription potentiel du gène 2, ou pour le moins, la probabilité pour que ces deux gènes soient impliqués dans une même voie métabolique est élevée.

L’existence d’une représentation selon un DBN défini par le DAG $\tilde{\mathcal{G}}$ repose uniquement sur les relations de dépendance entre les variables observées. Ainsi, modéliser un DBN selon le DAG $\tilde{\mathcal{G}}$ revient à considérer d’une part que le processus X est markovien d’ordre 1 et d’autre part que les variables observées sont simultanément indépendantes, c’est-à-dire que deux variables X_t^i et X_t^j observées au même instant t sont indépendantes sachant le passé du processus. L’unicité de ce DAG $\tilde{\mathcal{G}}$ repose essentiellement sur l’hypothèse que le processus X admet une densité par rapport à la mesure de Lebesgue sur $\mathbb{R}^{p \times n}$, c’est à dire d’une façon très simplifiée, que l’on peut considérer qu’il n’y a pas de variables rigoureusement identiques au sein du processus X .

Il s’agit ensuite de reconstruire la topologie de ce graphe alors que le nombre de mesures n est très inférieur au nombre p de gènes considérés. Je propose pour cela une méthode d’inférence originale qui repose sur la considération d’indépendances d’ordre partiel. Cette approche a été introduite par Wille et al. ([WZV⁺04], [WB06]) pour l’inférence de graphes non orientés dans le cadre de modèles gaussiens (GGM pour *Graphical Gaussian Model*). Wille et al. proposent ainsi d’approcher le graphe de concentration, qui représente les dépendances entre couple de variables conditionnellement à l’ensemble des variables observées, par le graphe des dépendances d’ordre 1. Castelo and Roverato [CR06] ont ensuite étendu ces résultats pour des graphes de dépendances partielles d’ordre q ($q \geq 1$). Ces approches permettent d’inférer un graphe non orienté ; je propose ici d’utiliser ce concept pour l’inférence de réseaux bayésiens dynamiques.

Ainsi, je définis le DAG $\mathcal{G}^{(q)}$ représentant les dépendances partielles d’ordre q entre les couples de variables successives au sein du processus temporel X . L’intérêt de ce graphe est de permettre d’approcher le DAG $\tilde{\mathcal{G}}$ tout en offrant l’avantage de pouvoir être estimé par des méthodes usuelles. En effet, pour de petites valeurs de q ($q \ll n$), on dispose de suffisamment de répétitions pour tester la nullité d’un coefficient de régression dans un modèle défini par q prédicteurs,

$$X_t^i = a_i + \sum_{j=1}^q b_{ij} X_{t-1}^j + e_t^i,$$

où e_t^i suit une loi gaussienne centrée d’écart-type σ^i . Il s’agit alors de caractériser ces DAG $\mathcal{G}^{(q)}$ dans le cadre des réseaux bayésiens dynamiques, et ce en particulier par rapport au “vrai” DAG $\tilde{\mathcal{G}}$

que l'on cherche à reconstruire. La théorie des modèles graphiques représentés par un DAG [Lau96] permet de montrer la capacité des DAG $\mathcal{G}^{(q)}$ à approcher $\tilde{\mathcal{G}}$ (voir section 3.3). Dans certains cas, ces deux graphes coïncident exactement. Pour le moins, sous l'hypothèse que le processus X est *faithful* au DAG $\tilde{\mathcal{G}}$ [SGS93], c'est-à-dire que $\tilde{\mathcal{G}}$ représente exactement l'ensemble des relations de dépendance entre les variables $\{X_t^i\}$, le DAG $\tilde{\mathcal{G}}$ est inclus dans $\mathcal{G}^{(1)}$.

Sur la base de ce résultat $\tilde{\mathcal{G}} \subseteq \mathcal{G}^{(1)}$, j'introduis une nouvelle méthode d'inférence de réseaux bayésiens dynamiques qui permet de faire face à la dimension du problème. Cette procédure consiste à inférer tout d'abord le graphe $\mathcal{G}^{(1)}$, ce qui permet de réduire le nombre d'interactions potentielles p^2 . En effet, étant donné que le DAG $\tilde{\mathcal{G}}$ contient un grand nombre p de gènes et que seul un petit nombre d'entre eux est impliqué dans un mécanisme de régulation, on s'attend à ce que le graphe $\tilde{\mathcal{G}}$ soit relativement "vide". Aussi, le DAG $\mathcal{G}^{(1)}$ qui représente les dépendances partielles d'ordre 1 du DAG $\tilde{\mathcal{G}}$ contient déjà très peu d'arêtes. Il est alors possible d'inférer dans un second temps le vrai DAG $\tilde{\mathcal{G}}$ à partir des arêtes de $\mathcal{G}^{(1)}$ au moyen de tests usuels. Cette procédure se révèle performante à la fois sur des données réelles (*S. cerevisiae*, *A. thaliana*) et simulées.

1.5.3 Reconstruire un réseau *chronologique* par MCMC à sauts réversibles

La définition du DAG $\tilde{\mathcal{G}}$ pour la modélisation de réseaux bayésiens dynamiques repose sur une hypothèse forte, celle de l'homogénéité des relations de dépendance au cours du temps. C'est-à-dire que, comme pour le graphe de la Figure 1.12, une arête est soit présente tout au long de la cinétique observée, soit absente. Or cette hypothèse n'est pas toujours vérifiée. On peut citer l'exemple très étudié du cycle cellulaire qui est composé de 4 phases mettant en jeu des facteurs de transcription différents. Les phases du cycle cellulaire sont bien connues et on peut dans ce cas envisager d'estimer un réseau pour chaque phase. Mais que faire dans le cas contraire ?

D'une manière générale, on peut s'attendre à ce que l'ensemble des actions de régulation ne soient pas toutes effectuées simultanément mais qu'au contraire elles se produisent successivement et selon une chronologie bien définie. C'est le cas notamment de la réaction au bénomyl chez la levure. Le bénomyl est un anti-mitotique qui inhibe la formation des microtubules. Suite à l'ajout de cette drogue, Lucau-Danila et al. [LDLK⁺05] ont mis en évidence une sur-expression rapide de certains gènes et plus tardive pour d'autres gènes. Il s'agit alors de délimiter la plage au cours de laquelle chaque relation de régulation est effective et ainsi de retrouver la chronologie des interactions. J'introduis pour cela dans le chapitre 4 un modèle de ruptures qui permet de décrire un réseau *chronologique*, c'est-à-dire un réseau qui varie au cours du temps. Ce modèle est schématisé Figure 1.13 dans le cas de 2 points de ruptures. Sur cet exemple, les relations de dépendances diffèrent selon les plages définies par le vecteur de ruptures (t_1, t_2, t_3, t_4) .

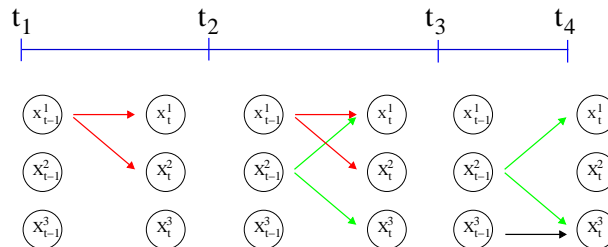


FIG. 1.13 – Réseau non-homogène à 2 instants de ruptures t_2 et t_3 . Les vecteurs de ruptures de chaque gène sont $\xi^1 = (t_1, t_2, t_3, t_4)$, $\xi^2 = (t_1, t_3, t_4)$ et $\xi^3 = (t_1, t_3, t_4)$.

Il s'agit alors d'estimer la dimension du modèle (nombre de points de ruptures, nombre d'arêtes pour chaque phase) ainsi que la valeur des paramètres. Très récemment, Rao et al. [RHISE07] ont

eux aussi proposé de reconstruire un réseau qui varie au cours du temps. Cependant, ils proposent de détecter dans un premier temps les points de rupture pour chacun des gènes observés ; puis de regrouper en “cluster” les gènes qui partagent les mêmes points de rupture et dont les niveaux d’expression ont un comportement similaire ; et enfin de reconstruire un réseau de dépendance entre les gènes d’un même cluster. Je propose dans le chapitre 4 une approche qui permet d’inférer *simultanément* la position des points de rupture pour chaque gène et la structure du réseau décrivant le processus de régulation dans chacune des plages.

J’introduis pour cela une procédure d’inférence fondée sur des méthodes de Monte Carlo Markov Chain (MCMC) à sauts réversibles, précisément introduites par [Gre95] pour faire face au cas où la dimension du modèle à estimer est inconnue. Grâce à l’imbrication de deux niveaux de MCMC à sauts réversibles, la détection des positions de rupture et de la structure du graphe décrivant les relations de dépendances au sein de chaque phase est effectuée de façon simultanée.

Ainsi, cette procédure est tout d’abord une méthode MCMC à sauts réversibles pour la détection des points de rupture. Elle est composée de quatre mouvements ; les trois premiers consistent à ajouter, éliminer, ou déplacer un point de rupture et le quatrième à modifier la structure du réseau défini pour chacune des plages délimitées par le vecteur de ruptures courant. Ce dernier mouvement est à nouveau une étape de MCMC à sauts réversibles qui permet d’ajouter une arête, d’en éliminer une ou simplement de modifier les paramètres du modèle décrivant chaque plage.

Cette procédure à deux niveaux de MCMC à sauts réversibles permet ainsi d’estimer simultanément la densité a posteriori des points de ruptures pour chaque gène, ainsi que celle de la structure du réseau décrivant chaque plage. L’approche est illustrée par une étude de la réponse au bénomyl chez la levure *S. cerevisiae*.

Chapitre 2

An EM algorithm for estimation in the Mixture Transition Distribution model

S. Lèbre and P.-Y. Bourguignon

This article is to appear in the Journal of Statistical Computation and Simulation.

Abstract

The Mixture Transition Distribution (MTD) model was introduced by Raftery to face the need for parsimony in the modeling of high-order Markov chains in discrete time. The particularity of this model comes from the fact that the effect of each lag upon the present is considered separately and additively, so that the number of parameters required is drastically reduced. However, the efficiency for the MTD parameter estimations proposed up to date still remains problematic on account of the large number of constraints on the parameters. In this paper, an iterative procedure, commonly known as Expectation-Maximization (EM) algorithm, is developed cooperating with the principle of Maximum Likelihood Estimation (MLE) to estimate the MTD parameters. Some applications of modeling MTD show the proposed EM algorithm is easier to be used than the algorithm developed by Berchtold. Moreover, the EM Estimations of parameters for high-order MTD models led on DNA sequences outperform the corresponding fully parametrized Markov chain in terms of Bayesian Information Criterion.

A software implementation of our algorithm is available in the library `seq++` at <http://stat.genopole.cnrs.fr/seqpp>.

Keywords: Markov chain; mixture transition distribution (MTD); Parsimony; Maximum likelihood; EM algorithm;

2.1 Introduction

While providing a useful framework for discrete-time sequence modeling, higher-order Markov chains suffer from the exponential growth of the parameter space dimension with respect to the order of the model, which results in the inefficiency of the parameters' estimations when a limited amount of data is available. This fact motivates the developments of approximate versions of higher-order Markov chains, such as the Mixture Transition Distribution (MTD) model [Raf85, BR02] and variable length Markov chains [BW99]. Thanks to a simple structure, where each lag contributes to the prediction of the current letter in a separate and additive way, the

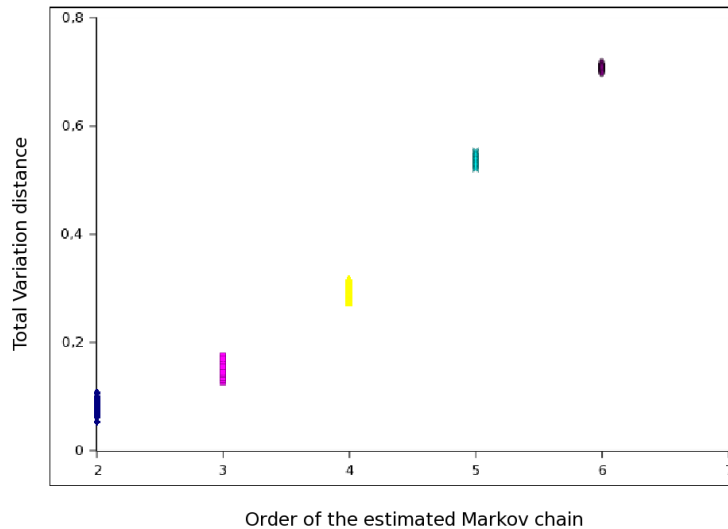


Figure 2.1: Total variation distance between distributions estimated from randomly generated sequences and the generating distribution. The generating model is of order 5, and the random sequences are 5000 letters long.

dimension of model parameter space grows only linearly with respect to the order of the MTD model.

Nevertheless, Maximum Likelihood Estimation (MLE) in the MTD model is subject to such constraints that analytical solutions are beyond the reach of present methods. One has thus to retort to numerical optimization procedures. The most powerful method proposed to this day is due to Berchtold [Ber01], and relies on an ad-hoc optimization method. In this paper, we propose to fit the MTD model into the general framework of hidden variable models, and derive a version of the classical EM algorithm for the estimations of its parameters.

In this first section, we define the MTD model and recall its main features and some of its variants. Parametrization of the model is discussed in section 2, where we establish that under the most general definition, it is not identifiable. Then we shed light on an identifiable set of parameters. Derivations of the update formulas involved by the EM algorithm are detailed in section 3. We finally illustrate our method by some applications to biological sequence modeling.

Need for parsimony Markov models are pertinent to analyze m -letter words' composition of a sequence of random variables [FG87, DEKM99]. Nevertheless, the length m of the words the model accounts for has to be chosen by the statistician. On the one hand, a high order is always preferred since it can capture strictly more information. On the other hand, since the parameter's dimension increases exponentially fast with respect to the order of the model, higher order models cannot be accurately estimated. Thus, a trade-off has to be drawn to optimize the amount of information extracted from the data.

We illustrate this phenomenon by running a simple experiment: by using a randomly chosen Markov chain transition matrix of order 5, we sample 1000 sequences of length 5000. Each of them is then used to estimate a Markov model transition matrix of order varying from 2 to 6. For each of these estimates, we have plotted the total variation distance with respect to the generating model (see Figure 2.1), computed as the quantity $D_{VT}(P, Q) = \sum_{x \in \mathcal{Y}^n} |P(x) - Q(x)|$

for distributions P and Q . It turns out that the optimal estimation in terms of total variation distance between genuine and estimated distributions is obtained with a model of order 2 whereas the generating model is of order 5.

Mixture Transition Distributions aim at providing a model accounting for the number of occurrences of m -letter words, while avoiding the exponential increase with respect to m of the full Markov model parameter's dimension (See Table 2.1 for a comparison of the models' dimensions).

MTD modeling Let $\mathbf{Y} = (Y_1, \dots, Y_n)$ be a sequence of random variables taking values in the finite set $\mathcal{Y} = \{1, \dots, q\}$. We use the notation,

$$\mathbf{Y}_{t_1}^{t_2} = (Y_{t_1}, Y_{t_1+1}, \dots, Y_{t_2})$$

to refer to the subsequence of the $t_2 - t_1 + 1$ successive variables. In the whole paper, vectors and matrices are denoted by bold letters.

Definition 1 *The random sequence \mathbf{Y} is said to be an m^{th} order MTD sequence if*

$$\begin{aligned} \forall t > m, \forall y_1, \dots, y_t \in \mathcal{Y}, \quad \mathbb{P}(Y_t = y_t | \mathbf{Y}_1^{t-1} = \mathbf{y}_1^{t-1}) &= \sum_{g=1}^m \varphi_g \mathbb{P}(Y_t = y_t | Y_{t-g} = y_{t-g}) \\ &= \sum_{g=1}^m \varphi_g \boldsymbol{\pi}_g(y_{t-g}, y_t). \end{aligned} \quad (2.1)$$

where the vector $\boldsymbol{\varphi} = (\varphi_1, \dots, \varphi_m)$ is subject to the constraints:

$$\forall g \in \{1, \dots, m\}, \quad \varphi_g \geq 0, \quad (2.2)$$

$$\sum_{g=1}^m \varphi_g = 1. \quad (2.3)$$

and the matrices $\{\boldsymbol{\pi}_g = [\mathbb{P}(Y_t = j | Y_{t-g} = i)]_{i,j \in \mathcal{Y}}; 1 \leq g \leq m\}$ are $q \times q$ stochastic matrices.

A m th-order MTD model is thus defined by a vector parameter,

$$\boldsymbol{\theta} = \left(\varphi_g, (\pi_g(i, j))_{i,j \in \mathcal{Y}} \right)_{1 \leq g \leq m}$$

which belongs to the space

$$\Theta = \left\{ \boldsymbol{\theta}; \forall 1 \leq g \leq m, 0 \leq \varphi_g \leq 1; \sum_{g=1}^m \varphi_g = 1; \right. \\ \left. \forall i, j \in \mathcal{Y}, 0 \leq \pi_g(i, j) \leq 1 \text{ and } \sum_{j \in \mathcal{Y}} \pi_g(i, j) = 1 \right\}.$$

It is obvious from the first equality in equation (2.1) that the MTD model fulfills the Markov property. Thus, MTD models are Markov models with the particularity that each lag Y_{t-1}, Y_{t-2}, \dots contributes additively to the distribution of the random variable Y_t . Berchtold and Raftery [BR02] published a complete review of the MTD model. They recall theoretical results on the limiting behavior of the model and on its auto-correlation structure. Details are given about several

extensions of this model, such as infinite-lag models, or infinite countable and continuous state space.

We have to point out that Raftery [Raf85] defined the original model with the same transition matrix $\boldsymbol{\pi}$ for each lag $\{Y_{t-g}\}_{g=1,\dots,m}$. In the sequel, we refer to this model as the *single* matrix MTD model. Later, Berchtold [Ber95] introduced a more general definition of the MTD models as a mixture of transitions from different subsets of lagged variables $\{Y_{t-m}, \dots, Y_{t-1}\}$ to the present one Y_t , eventually discarding some of the dependencies. In this paper, we focus on a slightly more restricted model having a specific but same order transition matrix $\boldsymbol{\pi}_g$ for each lag Y_{t-g} . We denote by MTD_l the MTD model which has a l -order transition matrix for each lag (Definition 2). From now on, the MTD model defined by (2.1) is denoted accordingly by MTD_1 .

Definition 2 *The random sequence \mathbf{Y} is a m -order MTD_l sequence if, for all $l, m \in \mathbb{N}$ such that $l < m$, and all $\mathbf{y}_1^t \in \mathcal{Y}^t$:*

$$\begin{aligned} \mathbb{P}(Y_t = y_t | \mathbf{Y}_1^{t-1} = \mathbf{y}_1^{t-1}) &= \mathbb{P}(Y_t = i_t | \mathbf{Y}_{t-m}^{t-1} = \mathbf{y}_{t-m}^{t-1}) \\ &= \sum_{g=1}^{m-l+1} \varphi_g \mathbb{P}(Y_t = y_t | \mathbf{Y}_{t-g-l+1}^{t-g} = \mathbf{y}_{t-g-l+1}^{t-g}) \\ &= \sum_{g=1}^{m-l+1} \varphi_g \boldsymbol{\pi}_g(\mathbf{y}_{t-g-l+1}^{t-g}, y_t). \end{aligned}$$

holds, where $\boldsymbol{\pi}_g$ is a $q^l \times q$ transition matrix.

Trade-off between dimension and maximal likelihood Even though MTD models involve a restricted amount of parameters compared to Markov chains, increasing the order l of the model may result in efficiency of the MLE decreased. The quality of the trade-off between goodness-of-fit and generalization error a model achieves can be assessed against classical model selection criteria, such as the Bayesian Information Criterion (see illustrations in section 2.4.2).

However, computing the BIC requires the knowledge of the dimension of the model. This dimension is usually computed as the dimension of the parameter space for a bijective parametrization. In the specific case of the MTD models, the original single-matrix model is parametrized in a bijective way, whereas its generalized version with specific transition matrices for each lag is over-parametrized: in appendix 2.5.1 is given an example of two distinct values of the parameters $(\boldsymbol{\varphi}, \boldsymbol{\pi})$, which both define the same MTD_1 distribution. The dimension of the model is thus lower than the dimension of the parameter space, and computing the BIC using the parameter space dimension would over-penalize the models. A tighter upper bound of the dimension of the MTD_l model is derived in section 2.2, a bound which is used later to compute the BIC.

The question of estimation As a counterpart for their parsimony, MTD parameters are difficult to be estimated due to the constraints that the transition probabilities $\{\mathbb{P}(i_m \dots i_1; i_0); i_m, \dots, i_0 \in \mathcal{Y}\}$ have to comply to. There is indeed no analytical solution to the maximization of the log-likelihood $L_y(\boldsymbol{\theta}) = \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{Y} = \mathbf{y})$ of the MTD models under the constraints the vector $\boldsymbol{\varphi}$ and the stochastic matrices $\boldsymbol{\pi}_g$ have to fulfill. For a given sequence $\mathbf{y} = y_1, \dots, y_n$ of length n , we recall that the loglikelihood of the sequence \mathbf{y} under the MTD_1 model writes

$$\begin{aligned} L_y(\boldsymbol{\theta}) &= \log \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{Y}_1^n = \mathbf{y}_1^n) \\ &= \log \left\{ \mathbb{P}(\mathbf{Y}_1^m = \mathbf{y}_1^m) \prod_{t=m+1}^n \left(\sum_{g=1}^m \varphi_g \boldsymbol{\pi}_g(y_{t-g}, y_t) \right) \right\}. \end{aligned}$$

-
- Compute partial derivatives of the log likelihood according to each element of the vector,
 - choose a value δ in $[0, 1]$,
 - add δ to the the component with the largest derivative, and subtract δ from the one with the smallest derivative.
-

Figure 2.2: Berchtold's Algorithm for MTD models

The estimation of the original single matrix MTD model already aroused a lot of interest. Although any distribution from this model is defined by a unique parameter θ , the maximum likelihood can not be analytically determined. Li and Kwok [LK90] propose an interesting alternative to the maximum likelihood with a minimum chi-square method. Nevertheless, they carry out estimations by using a non-linear optimization algorithm that is not explicitly described. Raftery and Tavaré [RT94] obtain approximations of both maximum likelihood and minimum chi-square estimates with numerical procedures from the NAG library which is not freely available. They also show that the MTD model can be estimated using GLIM (Generalized Linear Interactive Modeling) in the specific case where the state space's size q equals 2. Finally, Berchtold [Ber01] developed an ad hoc iterative method implementing a constrained gradient descent optimization. This algorithm is based on the assumption that the vector φ and each row of the matrix π are independent. It consists in successively updating each of these vectors constrained to have a sum of components equal to 1 as exposed in Figure 2.2.

This algorithm has been shown to perform at least better than the previous methods, and it can be extended to the case of the MTD_l models. In this latter case, it estimates *one* of the parameter vectors $\{(\varphi_g, \pi_g); 1 \leq g \leq m\}$ describing the maximum-likelihood MTD distribution. Nevertheless, the choice of the *alteration parameter* δ remains an issue of the method. An in-depth discussion of the strategy used to update the alteration parameter δ can be found in [Ber01].

We propose to approximate the maximum likelihood estimate $\{\hat{\mathbb{P}}_{ML}(i_m \dots i_1; i_0); i_m, \dots, i_0 \in \mathcal{Y}\}$ for MTD model by coming down to a better known problem: estimation of incomplete data with an Expectation-Maximization (EM) algorithm [DLR77]. We introduce a simple estimation method which allows to approximate *one* parameter vector $\theta = \{(\varphi_g, \pi_g); 1 \leq g \leq m\}$ maximizing the log-likelihood.

2.2 Upper bound of the MTD model dimension

The MTD_1 model is over-parametrized. We provide an example of two distinct parameter values (φ, π) defining the same 2^{nd} -order MTD_1 model in appendix 2.5.1. Moreover, we propose a new parameter set whose dimension is lower. It stems from the straightforward remark that the m th-order MTD_1 model satisfies the following proposition:

Proposition 1 *Transition probabilities of a m th-order MTD_1 model satisfy:*

$$\forall i_m, \dots, i_g, \dots, i_0, i'_g \in \mathcal{Y},$$

$$\mathbb{P}(i_m \dots i_g \dots i_1; i_0) - \mathbb{P}(i_m \dots i'_g \dots i_1; i_0) = \varphi_g [\pi_g(i_g, i_0) - \pi_g(i'_g, i_0)]. \quad (2.4)$$

This simply means that the left-hand side of equation (2.4) only depends on the parameter components associated to lag g .

Consider a given distribution from MTD_1 with parameter $(\varphi_g, \boldsymbol{\pi}_g)_{1 \leq g \leq m}$, and let u be an arbitrary element of \mathcal{Y} . Each transition probability $\mathbb{P}(i_m \dots i_1; i_0)$ may be written :

$$\mathbb{P}(i_m \dots i_1; i_0) = \sum_{g=1}^m \varphi_g [\boldsymbol{\pi}_g(i_g, i_0) - \boldsymbol{\pi}_g(u, i_0)] + \sum_{g=1}^m \varphi_g \boldsymbol{\pi}_g(u, i_0). \quad (2.5)$$

From Proposition 1, it follows that each term of the first sum $\varphi_g [\boldsymbol{\pi}_g(i_g, i_0) - \boldsymbol{\pi}_g(u, i_0)]$ equals the difference of probabilities $\mathbb{P}(u \dots u i_g u \dots u; i_0) - \mathbb{P}(u \dots u; i_0)$. The second sum is trivially the transition probability from the m -letter word $u \dots u$ to i_0 .

Let us denote the transition probabilities from m -letter words to the letter j , restricting to words differing from $u \dots u$ by at most one letter :

$$p_u(g; i, j) := \mathbb{P}(u \dots u i u \dots u; j), \quad (2.6)$$

where $u \dots u i u \dots u$ is the m -letter word whose letter in position g (from right to left) is i . The quantities in (2.6) are sufficient to describe the model, as stated in the following proposition.

Proposition 2 *The transition probabilities of a m th-order MTD_1 model satisfy:*

$$\forall u \in \mathcal{Y}, \forall i_m, \dots, i_g, \dots, i_0 \in \mathcal{Y},$$

$$\mathbb{P}(i_m, \dots, i_1; i_0) = \sum_{g=1}^m [p_u(g; i_g, i_0) - \frac{m-1}{m} p_u(i_0)].$$

where $p_u(j)$ denotes the value of $p_u(g; u, j)$, whatever the value of g .

For any arbitrary u element of \mathcal{Y} , a MTD_1 distribution can be parametrized by a vector θ_u from the $(q-1)[1+m(q-1)]$ -dimensional set $\bar{\Theta}_u$,

$$\bar{\Theta}_u = \left\{ \left((p_u(g; i, j))_{1 \leq g \leq m, i, j \in \mathcal{Y}} \text{ such that } \forall g \in \{1, \dots, m\}, \forall i \in \mathcal{Y}, \right. \right. \\ \left. \left. \sum_{j \in \mathcal{Y}} p_u(g; i, j) = 1 \text{ and } \forall g, g' \in \{1, \dots, m\}, p_u(g; u, j) = p_u(g'; u, j) \right\} \quad (2.7)$$

Note that not all θ_u in $\bar{\Theta}_u$ define a MTD_1 distribution: the sum $\sum_{g=1}^m p_u(g; i_g, i_0) - \frac{m-1}{m} p_u(i_0)$ may indeed fall outside the interval $[0, 1]$. For this reason, we can only claim that some subset Θ_u of $\bar{\Theta}_u$ is a parameter space for the MTD_1 model. However, as the components of a parameter $\theta_u \in \Theta_u$ are transition probabilities, two different parameter values can not define the same MTD distribution. The mapping of Θ_u on the MTD_1 model is thus bijective, which results in the dimension of $\bar{\Theta}_u$ being an upper bound of the dimension of the MTD model.

Whereas the original definition of the MTD_1 model (2.1) involves an $m-1+mq(q-1)$ -dimensional parameter set, this new parametrization lies in a smaller dimensional space, dropping $q(m-1)$ parameters.

Equivalent parametrization can be set for MTD models having higher order transition matrix for each lag. For any $l \geq 1$, a MTD_l model can be described by a vector composed of the transition probabilities $p_u^l(g; i_l \dots i_1, j) = \mathbb{P}(u \dots u i_l \dots i_1 u \dots u; j)$ for all l -letter words $i_l \dots i_1$. Denoting by Θ_u^l the corresponding parameter space, its dimension $|\Theta_u^l| = \sum_{k=2}^l [q^{k-2}(q-1)^3(m-k+1)] + (1+m(q-1))(q-1)$ is again much smaller than the number of parameters originally required to describe the MTD_l model (see [Gre05], section 2.2, for the counting details). A comparison of the dimensions according to both parametrizations appears in Table 2.1. We will now make use of the upper bound $|\Theta_u^l|$ of the model's dimension to penalize the likelihood in the assessment of MTD models goodness-of-fit (see section 2.4.2).

Table 2.1: **Number of independent parameters required to describe full Markov and MTD_l models (state space size: $q = 4$).** Except for the single matrix MTD model, MTD models originally defined with parameters $(\varphi, \boldsymbol{\pi})$ are over parametrized: the parameter $\boldsymbol{\theta}_u^l$, introduced in section 2.2, requires far less independent parameters. Note that the 1st order MTD₁ model (resp. 2nd order MTD₂ model) is equivalent to the 1st order (resp. 2nd order) full Markov model.

Order m	Full Markov	MTD ₁		MTD ₂	
		$ (\varphi, \boldsymbol{\pi}) $	$ \boldsymbol{\theta}_u^1 $	$ (\varphi, \boldsymbol{\pi}) $	$ \boldsymbol{\theta}_u^2 $
1	12	12	12		
2	48	25	21	48	48
3	192	38	30	97	84
4	768	51	39	146	120
5	3 072	64	48	195	156

2.3 EM Estimation

In this section, we expose an EM algorithm for the estimation of MTD models. Firstly, this procedure allows to maximize the likelihood without assuming the independence of parameters φ and $\boldsymbol{\pi}$ and offers the convergence properties of an EM algorithm. Secondly, from a technical point of view, the EM algorithm does not require any trick to fulfill the constraints holding on the $(\varphi, \boldsymbol{\pi})$ parameters as Berchtold's algorithm does. We expose here our estimation method of the MTD₁ model (2.1) having a specific 1st order transition matrix for each lag. The method can easily be adapted for single matrix MTD models as well as for MTD models having different types of transition matrix for each lag. Detailed derivations of the formulas for identical matrix MTD and MTD_l models are presented in appendix 2.5.2.

To estimate the transition probabilities $\{\mathbb{P}(i_m \dots i_1; i_0); i_m, \dots, i_0 \in \mathcal{Y}\}$ of a m th-order MTD₁ model, we propose to compute an approximation of *one* set of parameters $\boldsymbol{\theta} = (\varphi_g, \boldsymbol{\pi}_g)_{1 \leq g \leq m}$ which maximizes the likelihood.

2.3.1 Introduction of a hidden process

Our approach lies on a particular interpretation of the model. The definition of the MTD₁ model (2.1) is equivalent to a mixture of m hidden models where the random variable Y_t is predicted by one of the m Markov chains $\boldsymbol{\pi}_g$ with the corresponding probability φ_g . Indeed, the coefficients $(\varphi_g)_{g=1, \dots, m}$ define a probability measure on the finite set $\{1, \dots, m\}$ since they satisfy the constraints (2.2) and (2.3).

From now on, we consider a hidden state process S_1, \dots, S_n that determines the way according to which the prediction is carried out. The hidden state variables $\{S_t\}$, taking values in the finite set $\mathcal{S} = \{1, \dots, m\}$, are independent and identically distributed, with distribution

$$\forall t \leq n, \forall g \in \mathcal{S}, \quad \mathbb{P}(S_t = g) = \varphi_g.$$

The MTD₁ model is then defined as a hidden variable model. The observed variable Y_t depends on the current hidden state S_t and on the m previous variables Y_{t-1}, \dots, Y_{t-m} . This dependency structure of the model is represented as a Directed Acyclic Graph (DAG) in Figure 2.3. The hidden value at one position indicates which of those previous variables of transition matrices are

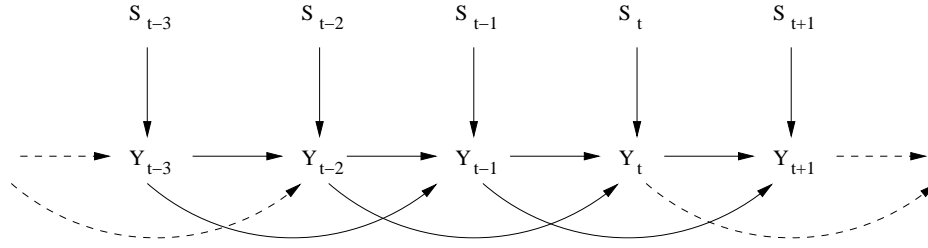


Figure 2.3: DAG dependency structure of a 2nd order MTD₁ model.

to be used to draw the current letter: conditional on the state S_t , the random variable Y_t only depends on the variable Y_{t-S_t} :

$$\forall t > m, \forall g \in \mathcal{S}, \quad \mathbb{P}(Y_t = y_t | Y_{t-m}^{t-1} = \mathbf{y}_{t-m}^{t-1}, S_t = g) = \pi_g(y_{t-g}, y_t).$$

So we carry out estimation in the MTD₁ models as estimation in a mixture model where the components of the mixture are m Markov chains, each one predicting the variable Y_t from one of the m previous variables.

2.3.2 EM algorithm

By considering a hidden variables model, we want to compute the maximum likelihood estimate from incomplete data. The EM algorithm introduced by Dempster et al. [DLR77] is a very classical framework for achieving such a task. It has proved to be particularly efficient at estimating various classes of hidden variable models. We make it entirely explicit in the case of the MTD models as summarized in Figure 2.4.

The purpose of the EM algorithm is to approximate the maximum of the log-likelihood of the incomplete data $L_y(\boldsymbol{\theta}) = \log \mathbb{P}_{\boldsymbol{\theta}}(Y = y)$ over $\boldsymbol{\theta} \in \Theta$ using the relationship

$$\forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta, L_y(\boldsymbol{\theta}) = Q(\boldsymbol{\theta} | \boldsymbol{\theta}') - H(\boldsymbol{\theta} | \boldsymbol{\theta}')$$

where the quantities Q and H are defined as follows :

$$\begin{aligned} Q(\boldsymbol{\theta} | \boldsymbol{\theta}') &= \mathbb{E} [\log \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{S}) | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}'] \\ H(\boldsymbol{\theta} | \boldsymbol{\theta}') &= \mathbb{E} [\log \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{S} | \mathbf{Y} = \mathbf{y}) | \mathbf{y}, \boldsymbol{\theta}'] \end{aligned}$$

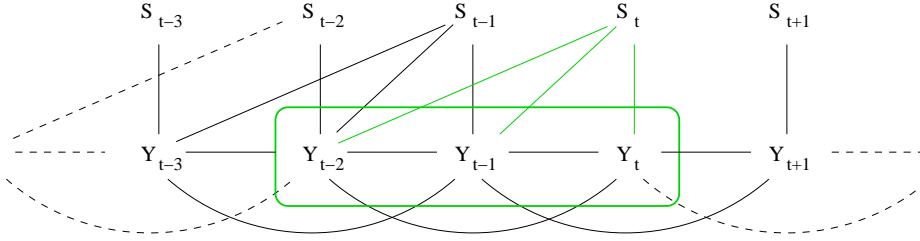
The EM algorithm is divided in two steps: E-step (Expectation) and M-step (Maximization). Both steps consist of, respectively, computing and maximizing the function $Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(k)})$, that is the log-likelihood of the complete model conditional on the observed sequence y and on the current parameter $\boldsymbol{\theta}^{(k)}$. Using the fact that the function $\boldsymbol{\theta} \rightarrow H(\boldsymbol{\theta} | \boldsymbol{\theta}^{(k)})$ is maximal in $\boldsymbol{\theta}^{(k)}$, Dempster et al. proved that this procedure necessarily increases the log-likelihood $L_y(\boldsymbol{\theta})$. See [Wu83] for a detailed study of the convergence properties of the EM algorithm.

We now derive analytical expressions for both E-step and M-step. In this particular case, the log-likelihood of the complete data $(\mathbf{Y}_{m+1}^n, \mathbf{S}_{m+1}^n)$ conditional on the first m observations \mathbf{Y}_1^m writes:

$$\begin{aligned} \log \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{Y}_{m+1}^n, \mathbf{S}_{m+1}^n | \mathbf{Y}_1^m) &= \sum_{t=m+1}^n \sum_{g=1}^m \sum_{i \in \mathcal{Y}} \sum_{j \in \mathcal{Y}} \mathbf{1}_{\{Y_{t-g}=i, Y_t=j, S_t=g\}} \log \pi_g(i, j) \\ &\quad + \sum_{t=m+1}^n \sum_{g=1}^m \mathbf{1}_{\{S_t=g\}} \log \varphi_g. \end{aligned} \quad (2.8)$$

-
- Compute the number of occurrences of each $(m + 1)$ -letters word $N(i_m^0)$,
 - Initialize parameters $(\varphi^{(0)}, \pi^{(0)})$,
 - Choose a stopping rule, *i.e.* an upper threshold ε on the increase of the log-likelihood,
 - Iterate E and M steps given by equations (2.13,2.14,2.15),
 - Stop when $L_y(\boldsymbol{\theta}^{(k+1)}) - L_y(\boldsymbol{\theta}^{(k)}) < \varepsilon$.
-

Figure 2.4: EM-Algorithm for MTD models

Figure 2.5: Moral graph of a 2^{nd} order MTD_1 model.

E-step The Estimation step is computing the expectation of this function (2.8) conditional on the observed data \mathbf{y} and the current parameter $\boldsymbol{\theta}^{(k)}$, that is calculating, for all $t > m$ and for all element g in $\{1, \dots, m\}$, the following quantity,

$$\mathbb{E}(\mathbf{1}_{\{S_t=g\}} | \mathbf{y}, \boldsymbol{\theta}^{(k)}) = \mathbb{P}(S_t = g | \mathbf{y}, \boldsymbol{\theta}^{(k)}). \quad (2.9)$$

Then, function Q writes:

$$Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(k)}) = \sum_{t=m+1}^n \sum_{g=1}^m \sum_{i \in \mathcal{Y}} \sum_{j \in \mathcal{Y}} [\mathbb{P}(S_t = g | \mathbf{y}, \boldsymbol{\theta}^{(k)}) \log \pi_g(i, j)] \mathbf{1}_{\{y_{t-g}=i, y_t=j\}} + \sum_{t=m+1}^n \sum_{g=1}^m \mathbb{P}(S_t = g | \mathbf{y}, \boldsymbol{\theta}^{(k)}) \log \varphi_g. \quad (2.10)$$

So E-step reduces to computing the probabilities (2.9), for which we derive an explicit expression by using the theory of graphical models in the particular case of DAG structured dependencies [Lau98]. First, remark that the state variable S_t depends on the sequence \mathbf{Y} only through the $m + 1$ variables $\{Y_{t-m}, \dots, Y_{t-1}, Y_t\}$:

$$\forall t \leq n, \forall g \in \{1, \dots, m\}, \quad \mathbb{P}(S_t = g | \mathbf{y}, \boldsymbol{\theta}) = \mathbb{P}(S_t = g | \mathbf{Y}_{t-m}^t = \mathbf{y}_{t-m}^t, \boldsymbol{\theta}). \quad (2.11)$$

Indeed, independence properties can be derived from the moral graph (Fig. 2.5) which is obtained from the DAG structure of the dependencies (Fig. 2.3) by “marrying” the parents, that is adding an edge between the common parents of each variable, and then deleting directions. In this moral graph, the set $\{Y_{t-m}, \dots, Y_t\}$ separates the variable S_t from the rest of the sequence $\{Y_1, \dots, Y_{t-m-1}\}$ so that applying corollary 3.23 from [Lau98] yields:

$$S_t \perp\!\!\!\perp (\mathbf{Y}_1^{t-m-1}, \mathbf{Y}_{t+1}^n) \mid \mathbf{Y}_{t-m}^t$$

From now on, we denote $\mathbf{i}_m^0 = i_m i_{m-1} \dots i_1 i_0$ any $(m + 1)$ -letter word composed of elements of \mathcal{Y} . For all g in $\{1, \dots, m\}$, for all \mathbf{i}_m^0 elements of \mathcal{Y} , Bayes’ Theorem gives:

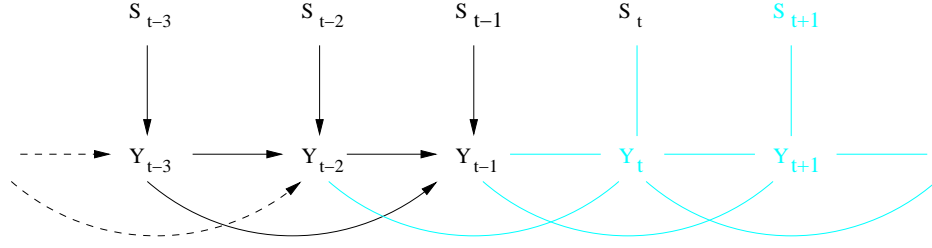


Figure 2.6: In black: graph of the smallest ancestral set containing S_t and the two variables (Y_{t-2}, Y_{t-1}) in the particular case of a 2^{nd} order MTD_1 model. (The part of the structure dependency DAG that is excluded from the smallest ancestral set appears here in light blue.)

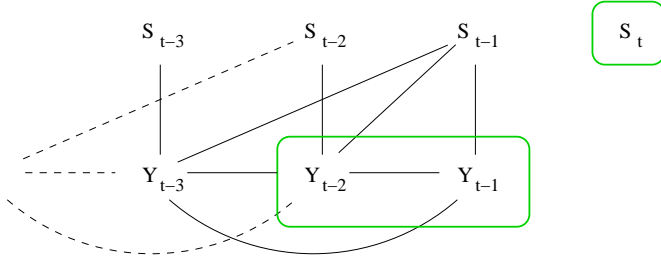


Figure 2.7: Moral graph of the smallest ancestral set in Figure 2.6. There is no path between S_t and the subset of 2 variables $\{Y_{t-2}, Y_{t-1}\}$.

$$\begin{aligned}
 & \mathbb{P}(S_t = g | Y_{t-m}^t = \mathbf{i}_m^0, \boldsymbol{\theta}) \\
 &= \frac{\mathbb{P}(S_t = g, Y_t = i_0 | \mathbf{Y}_{t-m}^{t-1} = \mathbf{i}_m^1, \boldsymbol{\theta})}{\mathbb{P}(Y_t = i_0 | \mathbf{Y}_{t-m}^{t-1} = \mathbf{i}_m^1, \boldsymbol{\theta})} \\
 &= \frac{\mathbb{P}(Y_t = i_0 | S_t = g, \mathbf{Y}_{t-m}^{t-1} = \mathbf{i}_m^1, \boldsymbol{\theta}) \mathbb{P}(S_t = g | \mathbf{Y}_{t-m}^{t-1} = \mathbf{i}_m^1, \boldsymbol{\theta})}{\sum_{l=1}^m \mathbb{P}(Y_t = i_0 | S_t = l, \mathbf{Y}_{t-m}^{t-1} = \mathbf{i}_m^1, \boldsymbol{\theta}) \mathbb{P}(S_t = l | \mathbf{Y}_{t-m}^{t-1} = \mathbf{i}_m^1, \boldsymbol{\theta})}. \tag{2.12}
 \end{aligned}$$

We show below that the probabilities $\mathbb{P}(Y_t = i_0 | S_t = g, \mathbf{Y}_{t-m}^{t-1} = \mathbf{i}_m^1, \boldsymbol{\theta})$ and $\mathbb{P}(S_t = g | \mathbf{Y}_{t-m}^{t-1} = \mathbf{i}_m^1, \boldsymbol{\theta})$ in expression (2.12) are entirely explicit. First, conditional on $\boldsymbol{\theta}$, the state S_t and the variables \mathbf{Y}_{t-m}^{t-1} , the distribution of Y_t writes:

$$\mathbb{P}(Y_t = i_0 | S_t = g, \mathbf{Y}_{t-m}^{t-1} = \mathbf{i}_m^1, \boldsymbol{\theta}) = \pi_g(i_g, i_0).$$

Second, although the state S_t depends on the $(m+1)$ -letter word \mathbf{Y}_{t-m}^t , which brings information about the probability of transition from \mathbf{Y}_{t-m}^{t-1} to Y_t , it does not depend on the m -letter word formed by the only variables \mathbf{Y}_{t-m}^{t-1} . This again follows from the same corollary in [Lau98]. The independence of the variables S_t and \mathbf{Y}_{t-m}^{t-1} is derived from the graph of the smallest ancestral set containing these variables, that is the subgraph containing S_t , \mathbf{Y}_{t-1}^{t-m} and the whole line of their ancestors (See Figure 2.6 for an illustration when $n = 2$). It turns out that, when considering the moralization of this subgraph (Figure 2.7), there is no path between S_t and the set \mathbf{Y}_{t-m}^{t-1} . This establishes $S_t \perp\!\!\!\perp \mathbf{Y}_{t-m}^{t-1}$ and we have

$$\mathbb{P}(S_t = g | \mathbf{Y}_{t-m}^{t-1} = \mathbf{i}_m^1, \boldsymbol{\theta}) = \mathbb{P}(S_t = g | \boldsymbol{\theta}) = \varphi_g.$$

Finally, the probability (2.12), is entirely determined by the current parameter $\boldsymbol{\theta}$ and does not depend on the time t .

Table 2.2: Maximum log-likelihood of MTD₁ models estimated by EM and Berchtold's algorithm (see [Ber01], section 5.1 and 6.2).

Order m	$q = \mathcal{Y} $	Berchtold	EM	Sequence
2	3	-486.4	-481.8	Pewee
	4	-1720.1	-1718.5	α A-Crystallin
3	3	-484.0	-480.0	Pewee
	4	-1710.6	-1707.9	α A-Crystallin

As a result, the k^{th} iteration of Estimation-step consists in calculating, for all g in $\{1, \dots, m\}$ and for all $m + 1$ -letters word \mathbf{i}_m^0 of elements of \mathcal{Y} ,

$$\forall g \in \{1, \dots, m\}, \forall i_m, \dots, i_1, i_0 \in \mathcal{Y},$$

$$\mathbb{P}_S^{(k)}(g|\mathbf{i}_m^0) = \mathbb{P}(S_t = g | \mathbf{Y}_{t-m}^t = \mathbf{i}_m^0, \boldsymbol{\theta}^{(k)}) = \frac{\varphi_g^{(k)} \pi_g^{(k)}(i_g, i_0)}{\sum_{l=1}^m \varphi_l^{(k)} \pi_l^{(k)}(i_l, i_0)}. \quad (2.13)$$

M-Step Maximization of the function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$ with respect to the constraints imposed on the vector $\boldsymbol{\varphi}$ and on the elements of the transition matrices $\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_m$ is easily achieved using Lagrange method:

$$\forall g \in \{1, \dots, m\}, \forall i, j \in \mathcal{Y},$$

$$\varphi_g^{(k+1)} = \frac{1}{n - m} \sum_{i_m \dots i_0} \mathbb{P}^{(k)}(g|\mathbf{i}_m^0) N(\mathbf{i}_m^0) \quad (2.14)$$

$$\boldsymbol{\pi}_g^{(k+1)}(i, j) = \frac{\sum_{i_m \dots i_{g+1} i_{g-1} \dots i_1} \mathbb{P}^{(k)}(g|\mathbf{i}_m^{g+1} \mathbf{i}_{g-1}^1 j) N(\mathbf{i}_m^{g+1} \mathbf{i}_{g-1}^1 j)}{\sum_{i_m \dots i_{g+1} i_{g-1} \dots i_1 i_0} \mathbb{P}^{(k)}(g|\mathbf{i}_m^{g+1} \mathbf{i}_{g-1}^0) N(\mathbf{i}_m^{g+1} \mathbf{i}_{g-1}^0)} \quad (2.15)$$

where sums are carried out for the variables $i_m, \dots, i_{g+1}, i_{g-1}, \dots, i_1, i_0$ taking values in \mathcal{Y} , n is the length of the observed sequence \mathbf{y} and $N(\mathbf{i}_m^0)$ the number of occurrences of the word \mathbf{i}_m^0 in this sequence.

Initialization To maximize the chance of reaching the global maximum, we run the algorithm from various starting points. One initialization is derived from contingency tables between each lag y_{t-g} and the present y_t as proposed by Berchtold [Ber01] and several others are randomly drawn from the uniform distribution.

A software implementation of our algorithm is available in the library seq++ at <http://stat.genopole.cnrs.fr/seqpp>.

2.4 Real data analysis

2.4.1 Comparison with Berchtold's Estimation

In this paper, we focus on estimation of the MTD _{l} model (see Definition 2) which has a specific but same order matrix transition for each lag. We evaluate the performance of the EM algorithm with comparison to the last and best algorithm to date, developed by Berchtold [Ber01]. Among

Figure 2.8: Estimation of a 2^{nd} order MTD_1 model on the song of the wood pewee. We use $u=1$ (song n°1) as reference letter to express the parameters defined in (2.7).

Estimates obtained with:

- Berchtold's algorithm ($L_y(\hat{\theta}) = -486.4$):

$$[\hat{p}_1(1; i, j)]_{1 \leq i, j \leq 3} = \begin{pmatrix} 0.754169 & 0.198791 & 0.073356 \\ 0.991696 & 0. & 0.03462 \\ 0.993579 & 0.003497 & 0.02924 \end{pmatrix}$$

$$[\hat{p}_1(2; i, j)]_{1 \leq i, j \leq 3} = \begin{pmatrix} 0.754169 & 0.198791 & 0.073356 \\ 0.137205 & 0.213411 & 0.649384 \\ 0.048023 & 0.927598 & 0.044116 \end{pmatrix}$$

- EM-algorithm ($L_y(\hat{\theta}) = -481.8$):

$$[\hat{p}_1(1; i, j)]_{1 \leq i, j \leq 3} = \begin{pmatrix} 0.75305 & 0.200475 & 0.046475 \\ 0.991475 & 0. & 0.008525 \\ 0.996425 & 0.003575 & 0. \end{pmatrix}$$

$$[\hat{p}_1(2; i, j)]_{1 \leq i, j \leq 3} = \begin{pmatrix} 0.75305 & 0.200475 & 0.046475 \\ 0.137525 & 0.21135 & 0.651125 \\ 0.02805 & 0.925475 & 0.046475 \end{pmatrix}$$

others, Berchtold estimates the parameters of MTD_l models on two sequences analyzed in previous articles: a time serie of the twilight song of the wood pewee and the mouse αA -Crystallin Gene sequence (the complete sequences appear in [RT94], Tables 7 and 12). The song of the wood pewee is a sequence composed of 3 distinct phrases (referred to as 1, 2, 3), whereas the αA -Crystallin Gene is composed of 4 nucleotides: a, c, g, t.

We apply our estimation method to these sequences and obtain comparable or higher value of the log-likelihood for both (see Tab. 2.2). Since the original parametrization of the MTD_1 model is not injective, it is not reasonable to compare their values. To overcome this problem, we computed the parameters from the set $\bar{\Theta}_u$ defined in (2.7). The estimated parameters (using a precision parameter $\varepsilon = 0.001$) of the 2^{nd} order MTD_1 model on the song of wood Pewee (first line of the Table 2.2) are exposed in Figure 2.8. Complete results appear in appendix 2.5.3, namely estimated parameters $\hat{\varphi}$, $\hat{\pi}_1$, $\hat{\pi}_2$ and their corresponding full 2^{nd} order transition matrices $\hat{\Pi}$.

For both sequences under study, Pewee and αA -crystallin, EM and Berchtold algorithms lead to comparable estimations. The EM algorithm proves here to be an effective method to maximize the log-likelihood of MTD models. Nevertheless, EM algorithm offers the advantage to be very easy to use. Whereas Berchtold's algorithm requires to set and update a parameter δ to alter the vector φ and each row of the matrices π_g , running the EM algorithm only requires the choice of the threshold ε in the stopping rule.

2.4.2 Estimation on DNA coding sequences

DNA coding regions are translated into proteins with respect to the genetic code, which is defined on blocks of three nucleotides called *codons*. Hence, the nucleotides in these regions are constrained in different ways according to their position in the codon. It is common in bioinformatics to use three different transition matrices to predict the nucleotides in the three positions of the codons. This model is called the *phased* Markov model.

Since we aim at comparing the goodness-of-fits of models with different dimensions, the maximal value of a penalized likelihood function against the dimension of parameter space will be used to assess each model. The Bayesian Information criterion [Sch78] for this evaluation is defined as:

$$BIC(\mathcal{M}) = -2L_y(\hat{\theta}_{\mathcal{M}}) + d(\mathcal{M}) \log n,$$

where $\hat{\theta}_{\mathcal{M}}$ stands for the maximum likelihood estimate of model \mathcal{M} . The lower the BIC a model achieves, the more pertinent it is.

BIC evaluation has been computed on DNA coding sequence sets from bacterial genomes. Each of these sequence sets has length ranging from 1 500 000 to 5 000 000. Displayed values in Figure 2.9 are averages over the 15 sequences set of the difference between the BIC value achieved by the full Markov model and the one achieved by the MTD model of the same order. Whenever this figure is positive, the MTD model has to be preferred to the full Markov model.

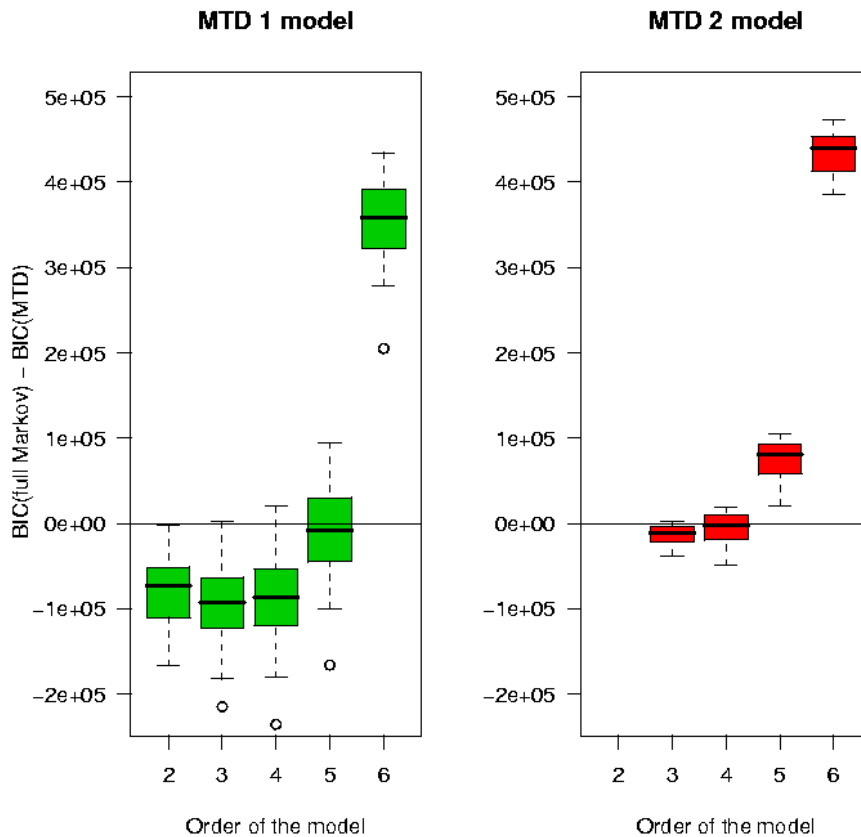


Figure 2.9: Difference according to the BIC criterion between MTD models and the corresponding fully parametrized Markov Model.

The full Markov model turns out to outperform the MTD_1 model when the order is inferior to 4. This is not surprising since the estimation is computed over large datasets that provide a sufficient amount of information with respect to the number of parameters of the full model. However, the 5th order MTD_1 model and full Markov model have comparable performances, and the MTD_1 model outperforms the full Markov model for higher orders. This is an evidence that although MTD_1 only approximate the full Markov models, their estimation accuracy decreases slower with the order.

Even more striking is the comparison of the MTD_2 model with the full Markov model. Whatever the order of the model, its goodness-of-fit is at least equivalent to the one achieved by the full Markov model. The MTD_l model turns out to be a satisfactory trade-off between dimension and estimation accuracy.

Acknowledgments

We thank Bernard Prum and Catherine Matias for their very constructive suggestions, and Vincent Miele for his implementation of the EM algorithm in the seq++ library. Moreover, we thank the referees for their comments and suggestions which improve this paper.

2.5 Appendix

2.5.1 Example of equivalent parameters defining the same MTD_1 model

Let the size state space be 4 as for DNA sequences $\mathcal{Y} = \{a, c, g, t\}$ and consider these two 2nd order MTD_1 model parameters $\boldsymbol{\theta}, \boldsymbol{\theta}'$.

$$\begin{aligned} \boldsymbol{\varphi} = (0.3, 0.7) \quad \boldsymbol{\pi}_1 = \begin{pmatrix} 0.1 & 0.2 & 0.3 & 0.4 \\ 0.4 & 0.3 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.2 & 0.4 \\ 0.4 & 0.2 & 0.2 & 0.2 \end{pmatrix} \quad \boldsymbol{\pi}_2 = \begin{pmatrix} 0.1 & 0.1 & 0.1 & 0.7 \\ 0.2 & 0.2 & 0.4 & 0.2 \\ 0.3 & 0.3 & 0.3 & 0.1 \\ 0.3 & 0.2 & 0.3 & 0.2 \end{pmatrix} \\ \boldsymbol{\varphi}' = (0.2, 0.8) \quad \boldsymbol{\pi}'_1 = \begin{pmatrix} 0.2 & 0.1 & 0.2 & 0.5 \\ 0.65 & 0.25 & 0.05 & 0.05 \\ 0.35 & 0.1 & 0.05 & 0.5 \\ 0.65 & 0.1 & 0.05 & 0.2 \end{pmatrix} \quad \boldsymbol{\pi}'_2 = \begin{pmatrix} 0.075 & 0.1375 & 0.15 & 0.6375 \\ 0.1625 & 0.225 & 0.4125 & 0.2 \\ 0.25 & 0.3125 & 0.325 & 0.1125 \\ 0.25 & 0.225 & 0.325 & 0.2 \end{pmatrix} \end{aligned}$$

Both parameters define the same 2nd order Markov transition matrix $\mathbf{\Pi}$.

$$\mathbf{\Pi} = \begin{matrix} & a & c & g & t \\ \begin{matrix} aa \\ ac \\ ag \\ at \\ ca \\ cc \\ cg \\ ct \\ ga \\ gc \\ gg \\ gt \\ ta \\ tc \\ tg \\ tt \end{matrix} & \begin{pmatrix} 0.1 & 0.13 & 0.16 & 0.61 \\ 0.19 & 0.16 & 0.13 & 0.52 \\ 0.13 & 0.13 & 0.13 & 0.61 \\ 0.19 & 0.13 & 0.13 & 0.55 \\ 0.17 & 0.2 & 0.37 & 0.26 \\ 0.26 & 0.23 & 0.34 & 0.17 \\ 0.2 & 0.2 & 0.34 & 0.26 \\ 0.26 & 0.2 & 0.34 & 0.2 \\ 0.24 & 0.27 & 0.3 & 0.19 \\ 0.33 & 0.3 & 0.27 & 0.1 \\ 0.27 & 0.27 & 0.27 & 0.19 \\ 0.33 & 0.27 & 0.27 & 0.13 \\ 0.24 & 0.2 & 0.3 & 0.26 \\ 0.33 & 0.23 & 0.27 & 0.17 \\ 0.27 & 0.2 & 0.27 & 0.26 \\ 0.33 & 0.2 & 0.27 & 0.2 \end{pmatrix} \end{matrix}$$

2.5.2 EM algorithm for other MTD models

Single matrix MTD model: iteration k .

E-Step $\forall g \in \{1, \dots, m\}, \forall i_m, \dots, i_1, i_0 \in \{1, \dots, q\}$,

$$\mathbb{P}_S^{(k)}(g|i_m^0) = \frac{\varphi_g^{(k)} \boldsymbol{\pi}^{(k)}(i_g, i_0)}{\sum_{l=1}^m \varphi_l^{(k)} \boldsymbol{\pi}^{(k)}(i_l, i_0)}.$$

M-Step $\forall g \in \{1, \dots, m\}, \forall i, j \in \{1, \dots, q\}$,

$$\begin{aligned} \varphi_g^{(k+1)} &= \frac{1}{n-m} \sum_{i_m \dots i_0} \mathbb{P}^{(k)}(g|i_m^0) N(\mathbf{i}_m^0) \\ \boldsymbol{\pi}^{(k+1)}(i, j) &= \frac{\sum_{g=1}^m \sum_{i_m \dots i_{g+1} i_{g-1} \dots i_1} \mathbb{P}^{(k)}(g|i_m^{g+1} \mathbf{i}_{g-1}^1 j) N(\mathbf{i}_m^{g+1} \mathbf{i}_{g-1}^1 j)}{\sum_{g=1}^m \sum_{i_m \dots i_{g+1} i_{g-1} \dots i_1 i_0} \mathbb{P}^{(k)}(g|i_m^{g+1} \mathbf{i}_{g-1}^0) N(\mathbf{i}_m^{g+1} \mathbf{i}_{g-1}^0)} \end{aligned}$$

where sums are carried out for the variables $i_m, \dots, i_{g+1}, i_{g-1}, \dots, i_1, i_0$ varying from 1 to q , n is the length of the observed sequence y and $N(\mathbf{i}_m^0)$ the number of occurrences of the word \mathbf{i}_m^0 in this sequence.

MTD_l model: iteration k.**E-Step** $\forall g \in \{1, \dots, m-l+1\}, \forall i_m, \dots, i_1, i_0 \in \{1, \dots, q\},$

$$\mathbb{P}_S^{(k)}(g|\mathbf{i}_m^0) = \frac{\varphi_g^{(k)} \boldsymbol{\pi}_g^{(k)}(\mathbf{i}_{g+l-1}^g, i_0)}{\sum_{h=1}^{m-l+1} \varphi_h^{(k)} \boldsymbol{\pi}_h^{(k)}(\mathbf{i}_{h+l-1}^h, i_0)}.$$

M-Step $\forall g \in \{1, \dots, m\}, \forall i_l, \dots, i_1, j \in \{1, \dots, q\},$

$$\begin{aligned} \varphi_g^{(k+1)} &= \frac{1}{n-m} \sum_{u_m \dots u_0} \mathbb{P}_S^{(k)}(g|\mathbf{u}_m^0) N(\mathbf{u}_m^0) \\ \boldsymbol{\pi}_g^{(k+1)}(i_l i_{l-1} \dots i_1, j) &= \frac{\sum_{u_m \dots u_{g+l} u_{g-1} \dots u_1} \mathbb{P}_S^{(k)}(g|\mathbf{u}_m^{g+l} \mathbf{i}_l^1 \mathbf{u}_{g-1}^1 j) N(\mathbf{u}_m^{g+l} \mathbf{i}_l^1 \mathbf{u}_{g-1}^1 j)}{\sum_{u_m \dots u_{g+l} u_{g-1} \dots u_1 u_0} \mathbb{P}_S^{(k)}(g|\mathbf{u}_m^{g+l} \mathbf{i}_l^1 \mathbf{u}_{g-1}^0) N(\mathbf{u}_m^{g+l} \mathbf{i}_l^1 \mathbf{u}_{g-1}^0)}, \end{aligned}$$

where sums are carried out for the variables $u_m, \dots, u_{g+l}, u_{g-1}, \dots, u_1, u_0$ varying from 1 to q , n is the length of the observed sequence y and $N(\mathbf{i}_m^0)$ the number of occurrences of the word \mathbf{i}_m^0 in this sequence.

2.5.3 2nd order MTD₁ estimates obtained on both the song of wood pewee and the mouse α A-Crystallin Gene sequence (Section 2.4.1).

1. Song of wood pewee

Berchtold's algorithm (see [Ber01], section 5.1): $L_y(\hat{\boldsymbol{\theta}}) = -486.4$.

$$\hat{\boldsymbol{\varphi}} = (0.269, 0.731) \quad \hat{\boldsymbol{\pi}}_1 = \begin{pmatrix} 0.097 & 0.739 & 0.164 \\ 0.980 & 0 & 0.020 \\ 0.987 & 0.013 & 0 \end{pmatrix} \quad \hat{\boldsymbol{\pi}}_2 = \begin{pmatrix} 0.996 & 0 & 0.004 \\ 0.152 & 0.020 & 0.828 \\ 0.003 & 0.997 & 0 \end{pmatrix}.$$

EM-algorithm: $L_y(\hat{\boldsymbol{\theta}}) = -481.8$.

$$\hat{\boldsymbol{\varphi}} = (0.275, 0.725) \quad \hat{\boldsymbol{\pi}}_1 = \begin{pmatrix} 0.102 & 0.729 & 0.169 \\ 0.969 & 0 & 0.031 \\ 0.987 & 0.013 & 0 \end{pmatrix} \quad \hat{\boldsymbol{\pi}}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0.151 & 0.015 & 0.834 \\ 0 & 1 & 0 \end{pmatrix}.$$

These estimated parameters define respectively the following 2nd order Markov transition matrices $\hat{\boldsymbol{\Pi}}_B$ and $\hat{\boldsymbol{\Pi}}_{EM}$.

$$\hat{\Pi}_B = \begin{pmatrix} 0.754169 & 0.198791 & 0.047040 \\ 0.991696 & 0. & 0.008304 \\ 0.993579 & 0.003497 & 0.02924 \\ 0.137205 & 0.213411 & 0.649384 \\ 0.374732 & 0.01462 & 0.610648 \\ 0.376615 & 0.018117 & 0.605268 \\ 0.028286 & 0.927598 & 0.044116 \\ 0.265813 & 0.728807 & 0.00538 \\ 0.267696 & 0.732304 & 0. \end{pmatrix} \quad \hat{\Pi}_{EM} = \begin{pmatrix} 0.75305 & 0.200475 & 0.046475 \\ 0.991475 & 0. & 0.008525 \\ 0.996425 & 0.003575 & 0. \\ 0.137525 & 0.21135 & 0.651125 \\ 0.37595 & 0.010875 & 0.613175 \\ 0.3809 & 0.01445 & 0.60465 \\ 0.02805 & 0.925475 & 0.046475 \\ 0.266475 & 0.725 & 0.008525 \\ 0.271425 & 0.728575 & 0. \end{pmatrix}$$

2. Mouse α A-Crystallin Gene sequence

EM-algorithm: $L_y(\hat{\theta}) = -1718.5$.

$$\hat{\varphi} = (0.562, 0.438),$$

$$\hat{\pi}_1 = \begin{pmatrix} 0.225 & 0.140 & 0.506 & 0.129 \\ 0.354 & 0.300 & 0.008 & 0.338 \\ 0.271 & 0.123 & 0.456 & 0.150 \\ 0.166 & 0.191 & 0.430 & 0.213 \end{pmatrix} \quad \hat{\pi}_2 = \begin{pmatrix} 0.094 & 0.600 & 0.149 & 0.157 \\ 0.335 & 0.271 & 0.153 & 0.241 \\ 0.185 & 0.415 & 0.099 & 0.301 \\ 0.192 & 0.370 & 0.129 & 0.309 \end{pmatrix}.$$

These estimated parameters define respectively the following 2^{nd} order Markov transition matrix $\hat{\Pi}_{EM}$.

$$\hat{\Pi}_{EM} = \begin{pmatrix} 0.167622 & 0.341480 & 0.349634 & 0.141264 \\ 0.240120 & 0.431400 & 0.069758 & 0.258722 \\ 0.193474 & 0.331926 & 0.321534 & 0.153066 \\ 0.134464 & 0.370142 & 0.306922 & 0.188472 \\ 0.273180 & 0.197378 & 0.351386 & 0.178056 \\ 0.345678 & 0.287298 & 0.071510 & 0.295514 \\ 0.299032 & 0.187824 & 0.323286 & 0.189858 \\ 0.240022 & 0.226040 & 0.308674 & 0.225264 \\ 0.207480 & 0.260450 & 0.327734 & 0.204336 \\ 0.279978 & 0.350370 & 0.047858 & 0.321794 \\ 0.233332 & 0.250896 & 0.299634 & 0.216138 \\ 0.174322 & 0.289112 & 0.285022 & 0.251544 \\ 0.210546 & 0.240740 & 0.340874 & 0.207840 \\ 0.283044 & 0.330660 & 0.060998 & 0.325298 \\ 0.236398 & 0.231186 & 0.312774 & 0.219642 \\ 0.177388 & 0.269402 & 0.298162 & 0.255048 \end{pmatrix}$$

No detail on the 2^{nd} order MTD₁ estimates from the mouse α A-Crystallin Gene sequence is given in [Ber01].

Chapter 3

Inferring dynamic genetic networks with low order independencies

S. Lèbre.

This article is submitted to the journal Statistical Application for Genetic and Molecular Biology.

Abstract

In this paper, we propose a novel inference method for dynamic genetic networks which makes it possible to face with a number of time measurements n much smaller than the number of genes p . The approach is based on the concept of low order conditional dependence graph that we extend here in the particular case of Dynamic Bayesian Networks. Most of our results are based on the theory of graphical models associated with the Directed Acyclic Graphs (DAGs). In this way, we define a DAG $\tilde{\mathcal{G}}$ which describes exactly the *full order conditional dependencies* given the past of the process. Then, to face with the large p and small n estimation case, we propose to approximate DAG $\tilde{\mathcal{G}}$ by considering low order conditional independencies. We introduce partial q^{th} order conditional dependence DAGs and analyze their probabilistic properties. In general, DAGs $\mathcal{G}^{(q)}$ differ from $\tilde{\mathcal{G}}$ but still reflect relevant dependence facts for sparse networks such as genetic networks. By using this approximation, we set out a non-bayesian inference method and demonstrate the effectiveness of this approach on both simulated and real data analysis. The inference procedure is implemented in the R package 'G1DBN' which is available from the CRAN archive.

Keywords: conditional independence, Bayesian networks, directed acyclic graphs, dynamic networks inference, time series modeling.

3.1 Introduction

The development of microarray technology allows to simultaneously measure the expression levels of many genes at a precise time point. Thus it has become possible to observe gene expression levels across a whole process like cell cycle or response to radiation or several treatments. The objective is now to recover gene regulation phenomena from this data. We are looking for simple relationships such as "gene i activates gene j ". But we also want to capture more complex scenarios such as auto-regulations, feed-forward loops, multi-component loops... as described by Lee et al. [LRR⁺02] in the transcriptional regulatory network of the yeast *Saccharomyces cerevisiae*.

To such an aim, we both need to accurately take into account temporal dependencies and to face with the dimension of the problem as the number p of observed genes is much higher than the number n of observation time points. Moreover we know that most of the observed genes are not taking part to the evolution of the system. So we want to determinate which are the few "active" agents, that are the agents being responsible for the evolution of the system and what are the relationships between them. In short, we want to infer a network representing the dependence relationships which govern a multiple elements-system from the observation of this system across short time series.

Such gene networks were firstly described by using static modeling and mainly non oriented networks. One of the first tools used to describe interaction between genes is the *relevance network* [BTS⁺00] or *correlation network* [SKFW03]. Better known as *covariance graph* [CW96] in the graphical models theory, this non directed graph describes the pair-wise correlation between genes. Its topology is derived from the covariance matrix between the gene expression levels; an undirected edge is drawn between two variables whenever they are correlated. Nevertheless, the correlation between two variables may come from the linkage with other variables. This creates spurious edges due to indirect dependence relationships.

Consequently, great interest has been taken in the *concentration graph* [Lau96], also called *covariance selection* model, which describes the *conditional* dependence structure between gene expression in Graphical Gaussian Models (GGMs). Let $Y = (Y^i)_{1 \leq i \leq p}$ be a multivariate Gaussian vector representing the expression levels of p genes. An undirected edge is drawn between two variables Y^i and Y^j whenever they are conditionally dependent given the remaining variables. The standard theory of estimation in GGMs [Whi90], [Lau96] can be exploited only when the number of measurements n is much higher than the number of variables p . This ensures that the sample covariance matrix is positive definite with probability one. Nevertheless, in most of the microarray gene expression data, we have to cope with the opposite situation ($n \ll p$). Thus, the growing interest for 'small n , large p ' furthered the development of numerous alternatives (Schäfer and Strimmer [SS05a] [SS05b], Waddell and Kishino [WK00b] [WK00a], Toh and Horimoto [TH02a] [TH02b], Wu et al. [WYS03], Wang et al. [WMH03]). Even though concentration graphs allow to point out some dependence relationships between genes, they do not offer an accurate description of the interactions. Firstly, no direction is given to the interactions. Secondly, some motifs containing cycles cannot be properly represented (see Figure 3.1).



Figure 3.1: A biological regulation motif (left) and the corresponding concentration graph (right). For all $i \geq 3$, Y^i is a Gaussian variable representing the expression level of gene G_i . Some cycles cannot be represented on the concentration graph.

Contrary to the previous undirected graphs, Bayesian networks (BNs) [FLNP00] model directed relationships. Based on a probabilistic measure, a BN representation of a model is defined by a Directed Acyclic Graph (DAG) and the set of conditional probability distributions of each variable given its parents in the DAG [Pea88]. Then the theory of graphical models [Whi90, Edw95, Lau96] allows to derive conditional independencies from this DAG. However the acyclicity constraint in static BNs is a serious restriction given the expected structure of genetic networks.

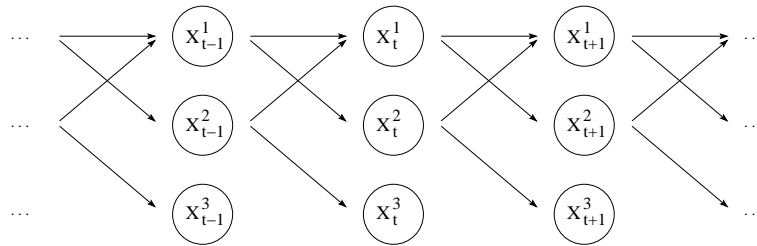


Figure 3.2: Dynamic network equivalent to the regulation motif in Figure 3.1 (left). Each vertex X_t^i represents the expression level of gene G_i at time t . This graph is acyclic and allows to define a Bayesian network.

Here comes the interest of Dynamic Bayesian networks (DBNs) first introduced for the analysis of gene expression time series by Friedman et al. [FMR98] and Murphy and Mian [MM99]. In DBNs, a gene is not anymore represented by a single vertex but by as much vertices as time points in the experiment. A dynamic network (Figure 3.2) can then be obtained by *unfolding in time* the initial cyclic motif in Figure 3.1 (left). The directions according to the time guarantees the acyclicity of this dynamic network and consequently allows to define a Bayesian network. The signs +/- showing the type of regulation in the biological motif do not appear in this DAG but they can be derived from model parameters estimates.

The very high number p of genes simultaneously observed raises a dimension problem. Moreover, a large majority of time series gene expression data contain no or very few repeated measurement(s) of the expression level of the same gene at a given time. Hence, we assume that the process is *homogeneous* across time. This consists of considering that the system is governed by the same rules during the whole experiment. Consequently, the temporal dependencies are homogeneous: any edge is present during the whole process. This is a strong assumption which is not necessarily satisfied. Nevertheless, this condition is necessary to carry out estimation. Indeed, in that case, we observe $n - 1$ repeated measurements of the expression level of each gene at two successive time points.

Up to now, various DBN representations based on different probabilistic models have been proposed (discrete models [OGP02, ZC05], multivariate auto-regressive process [ORS07], State Space or Hidden Markov Models [PRM⁺03, WZK04, RAG⁺04, BFG⁺05], nonparametric additive regression model [IGM02, IKG⁺03, KIM04, SI04]). See also Kim et al. [KIM03] for a review of such models. Facing with as much diversity, we expose here sufficient condition such that a model admits a DBN representation and we set out a straight interpretation in terms of dependencies between variables by using the theory of graphical models for DAGs. Our DBN representation is based on a DAG $\tilde{\mathcal{G}}$ (e.g. like the DAG of Fig. 3.2) which describes exactly the full order conditional dependencies given all the remaining *past* variables (see section 3.2). This approach extends the principle of the concentration graph showing conditional independencies to the dynamic case.

Even under homogeneity assumption, which enables to use the different time points as repeated measurements of the same process, we still have to deal with the 'curse of dimension' to infer the structure of DAG $\tilde{\mathcal{G}}$. The difficulty lies in facing with the large p and small n estimation case. Several inference methods have been proposed for the estimation of the topology of the various graphs quoted above. Among others, Murphy [Mur01] implemented several Bayesian structure learning procedures for dynamic models in the open-source Matlab package BNT (Bayes Net Toolbox); Ong et al. [OGP02] reduce the dimension of the problem by considering prior

knowledge; Perrin et al. [PRM⁺03] use an extension of the linear regression; Wu et al. [WZK04] use factor analysis and Beal et al. [BFG⁺05] develop a variational Bayesian method; Zou and Conzen [ZC05] limit potential regulators to the genes with either earlier or simultaneous expression changes and estimate the transcription time lag; Opgen-Rhein and Strimmer [ORS07] recently proposed a model selection procedure based on an analytic shrinkage approach. However, a powerful approach based on the consideration of zero- and first-order conditional independencies recently gained attention to model concentration graphs. When $n \ll p$, Wille et al. [WZV⁺04, WB06] propose to approximate the concentration graph by the graph \mathcal{G}_{0-1} describing zero- and first-order conditional independence. An edge between the variables Y^i and Y^j is drawn in the graph \mathcal{G}_{0-1} if and only if, zero- and first-order correlations between these two variables both differ from zero, that is, if the next conditions are satisfied,

$$\text{Corr}(Y^i, Y^j) \neq 0 \quad \text{and} \quad \forall k \in \{1, \dots, p\} \setminus \{i, j\}, \text{Corr}(Y^i, Y^j | Y^k) \neq 0, \quad (3.1)$$

where $\text{Corr}(Y^i, Y^j | Y^k)$ is the partial correlation between Y^i and Y^j given Y^k . Hence, whenever the possible correlation between two variables Y^i and Y^j can be entirely explained by the effect of some variable Y^k , no edge is drawn between them.

This procedure allows a drastic dimension reduction: by using first order conditional correlations, estimation can be carried out accurately even with a small number of observations. Even if the graph of zero- and first-order conditional independence differs from the concentration graph in general, it still reflects some measure of conditional independence. Wille et al. show through simulations that the graph \mathcal{G}_{0-1} offers a good approximation of sparse concentration graphs and demonstrate that both graphs even coincide exactly if the concentration graph is a forest ([WB06], Corollary 1). This approach has also been used by Magwene and Kim [MK04] and de la Fuente et al. [DIFBHM04] for estimating non-directed gene networks from microarray gene expression of the yeast *Saccharomyces cerevisiae*. Castelo and Roverato [CR06] investigate such non directed q^{th} order partial independence graphs for $q \geq 1$ and expose a sharp analysis of their properties. In this paper, we extend this approach by defining q^{th} order order conditional dependence DAGs $\mathcal{G}^{(q)}$ for DBN representations. Then, by basing on our results on these low order conditional dependence DAGs, we propose a novel inference method for dynamic genetic networks which makes it possible to face with the 'small n , large p ' estimation case.

The remainder of the paper is organized as follows. In section 3.2, we expose sufficient conditions for a DBN modeling of time series describing temporal dependencies. We notably show the existence of a minimal DAG $\tilde{\mathcal{G}}$ which allows such a DBN representation. To reduce the dimension of the estimation of the topology of $\tilde{\mathcal{G}}$, we propose to approximate $\tilde{\mathcal{G}}$ by q^{th} order conditional dependence DAGs $\mathcal{G}^{(q)}$ and analyze their probabilistic properties in section 3.3. From conditions on the topology of $\tilde{\mathcal{G}}$ and faithfulness assumption, we establish inclusion relationships between both DAGs $\tilde{\mathcal{G}}$ and $\mathcal{G}^{(q)}$. In section 3.4, we exploit our results on DAGs $\mathcal{G}^{(q)}$ to develop a non-Bayesian estimation procedure. Finally, validation is obtained on both simulated and real data in section 3.5. We notably expose our results for the analysis of two microarray time course data sets: the Spellman's yeast cell cycle data [SSZ⁺98] and the diurnal cycle data on the starch metabolism of *Arabidopsis Thaliana* collected by Smith et al. [SFC⁺04].

3.2 A DBN representation

Let $P = \{1 \leq i \leq p\}$ describe the set of observed genes and $N = \{1 \leq t \leq n\}$ the space of observation times. In this paper, we consider a discrete-time stochastic process $X = \{X_t^i; i \in P, t \in N\}$ taking real values and assume the joint probability distribution \mathbb{P} of the process X has

density f with respect to Lebesgue measure on $\mathbb{R}^{p \times n}$. We denote by $X_t = \{X_t^i; i \in P\}$ the set of the p random variables observed at time t and $X_{1:t} = \{X_s^i; i \in P, s \leq t\}$ the set of the random variables observed before time t .

In this section, we expose sufficient conditions under which the probability distribution \mathbb{P} admits a BN representation according to a dynamic network (e.g. in Figure 3.2). The main result is set out in Proposition 5; we show that it exists a BN representation according to a minimal DAG $\tilde{\mathcal{G}}$ whose edges describe exactly the set of direct dependencies between successive variables X_{t-1}^j, X_t^i given the past of the process. For an illustration, minimal DAG $\tilde{\mathcal{G}}_{AR(1)}$ is given in the particular case of an AR(1) model in subsection 3.2.4. The main interest of a DBN representation is to derive conditional dependence relationships between the variables by using the graphical theory associated with the DAGs. Note that, even though we need to consider a homogeneous DBN for the inference of gene interaction networks, the general framework (sections 3.2 and 3.3) is developed without assuming homogeneity.

Table 3.1: Notations

$P = \{1 \leq i \leq p\}$	set of observed genes,
$P_i = P \setminus \{i\}$	
$N = \{1 \leq t \leq n\}$	time space,
$X = \{X_t^i; i \in P, t \in N\}$	stochastic process (gene expression levels time series),
$\mathcal{G} = (X, E(\mathcal{G}))$	a DAG whose vertices are defined by X and edges by $E(\mathcal{G}) \subseteq X \times X$,
$\tilde{\mathcal{G}}$	the "true" DAG describing full order conditional dependencies,
$\mathcal{G}^{(q)}$	q^{th} order conditional dependence DAG,

3.2.1 Backgrounds

Let $\mathcal{G} = (X, E(\mathcal{G}))$ be a DAG whose vertices are the variables $X = \{X_t^i; i \in P, t \in N\}$ and whose set of edges $E(\mathcal{G})$ is a subset of $X \times X$. We quickly recall here elements of the theory of graphical models associated with the DAGs [Lau96].

Definition 3 *The parents of a vertex X_t^i in \mathcal{G} , denoted by $pa(X_t^i, \mathcal{G})$, are the variables having an edge pointing towards the vertex X_t^i in \mathcal{G} ,*

$$pa(X_t^i, \mathcal{G}) := \{X_s^j \text{ such that } (X_s^j, X_t^i) \in E(\mathcal{G}); j \in P, s \in N\}.$$

Proposition 3 (BN representation [Pea88]) *The probability distribution \mathbb{P} of the process X admits a Bayesian Network representation according to DAG \mathcal{G} whenever its density f factorizes as a product of the conditional density of each variable X_t^i given its parents in \mathcal{G} ,*

$$f(X) = \prod_{i \in P} \prod_{t \in N} f(X_t^i | pa(X_t^i, \mathcal{G})).$$

Definition 4 (Moral graph) *The moral graph \mathcal{G}^m of DAG \mathcal{G} is obtained from \mathcal{G} by first 'marrying' the parents (draw an undirected edge between each pair of parents of each variable X_t^i) and then deleting directions of the original edges of \mathcal{G} .*

For an illustration, Figure 3.3 exposes the moral graph of the DAG in Figure 3.2.

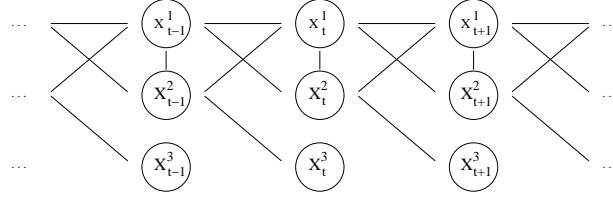


Figure 3.3: Moral graph of the DAG in Figure 3.2. For all $t > 1$, the parents of the variable X_t^1 are “married”, that is connected by a non directed edge.

Definition 5 (Ancestral set) *The subset S is ancestral if and only if, for all $\alpha \in S$, the parents of α satisfy $pa(\alpha, \mathcal{G}) \subseteq S$. Hence, for any subset S of vertices, there is a smallest ancestral set containing S which is denoted by $An(S)$. Then $\mathcal{G}_{An(S)}$ refers to the graph of the smallest ancestral set $An(S)$. See Figure 3.4 for an illustration.*

Throughout this paper, a central notion is that of conditional independence of random variables. Let $\mathbb{P}_{U,V,W}$ be the joint distribution of three random variables (U, V, W) . We say that U is *conditionally independent of V given W under $\mathbb{P}_{U,V,W}$* and write $U \perp\!\!\!\perp V \mid W$ whenever the variable U does not depend on V when considering the joint distribution $\mathbb{P}_{U,V,W}$. This result generalizes to sets of disjoint variables. Such conditional independence relationships can be set from a BN representation by using the graphical theory associated with the DAGs. Most of the results are based on the next proposition which is derived from the Directed global Markov property [Lau96].

Proposition 4 (Lauritzen [Lau96], Corollary 3.23) *Let \mathbb{P} admit a BN representation according to \mathcal{G} . Then,*

$$E \perp\!\!\!\perp F \mid S,$$

*whenever all paths from E to F intersect S in $(\mathcal{G}_{An(E \cup F \cup S)})^m$, the moral graph of the smallest ancestral set containing $E \cup F \cup S$. We say that S **separates** E from F .*

3.2.2 Sufficient conditions for a DBN representation

Assumption 1 *The stochastic process X_t is first-order Markovian,*

$$\forall t \geq 3, \quad X_t \perp\!\!\!\perp X_{1:t-2} \mid X_{t-1}.$$

Assumption 2 *For all $t \geq 1$, the random variables $\{X_t^i\}_{i \in P}$ are conditionally independent given the past of the process $X_{1:t-1}$, that is,*

$$\forall t \geq 1, \forall i \neq j, \quad X_t^i \perp\!\!\!\perp X_t^j \mid X_{1:t-1}.$$

We first assume that the observed process X_t is first-order Markovian (Assumption 1). That is the expression level of a gene at given time t only depends on the past through the expression level at the previous time $t - 1$. Then we assume that the variables observed simultaneously are conditionally independent given the past of the process (Assumption 2). In other words, we consider that time measurements are close enough so that a gene expression level X_t^i measured at time t is better explained by the previous time expression levels X_{t-1} than by some current expression level X_t^j .

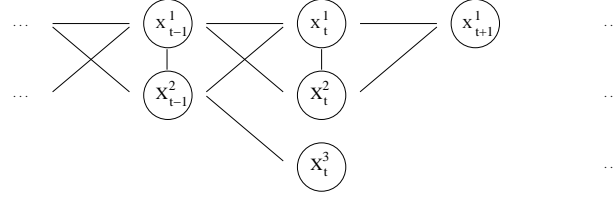


Figure 3.4: Moral graph of the smallest ancestral set containing the variables X_{t+1}^1 , its parents in the DAG in Figure 3.2 and X_t^2 . As the set (X_t^1, X_t^2) blocks all paths between X_t^3 and X_{t+1}^1 , we have $X_{t+1}^1 \perp\!\!\!\perp X_t^3 \mid (X_t^1, X_t^2)$.

From Assumptions 1 and 2, we establish in Lemma 1 the existence of a DBN representation of the distribution \mathbb{P} according to DAG \mathcal{G}_{full} which contains all the edges pointing out from a variable observed at some time $t - 1$ towards a variable observed at next time t . The direction of the edges according to the time guarantees the acyclicity of \mathcal{G}_{full} .

Lemma 1 *Under Assumptions 1 and 2, the probability distribution \mathbb{P} admits a DBN representation according to a DAG whose edges only join nodes representing variables observed at two successive time points, at least according to the DAG $\mathcal{G}_{full} = (X, \{(X_{t-1}^j, X_t^i)\}_{i,j \in P, t > 1})$ which has edges between any pair of successive variables.*

Proof. From assumption 1, the density f of the joint probability distribution of the process X writes as the product of conditional densities,

$$f(X) = f(X_1) \prod_{t=2}^n f(X_t | X_{t-1}), \quad (3.2)$$

where $f(X_t | X_{t-1})$ refers to the density of the conditional probability distribution of X_t given X_{t-1} .

From Assumption 2, for all $t > 1$, the conditional density $f(X_t | X_{t-1})$ writes as the product of the conditional density of each variable X_t^i given the set of variables X_{t-1} observed at the previous time,

$$f(X_t | X_{t-1}) = \prod_{i \in P} f(X_t^i | X_{t-1}). \quad (3.3)$$

From equations (3.2) and (3.3), the density f writes as the product of the conditional density of each variable X_t^i given its parents in \mathcal{G}_{full} . From Proposition 3, the probability distribution \mathbb{P} admits a BN representation according to \mathcal{G}_{full} . ■

3.2.3 Minimal DAG $\tilde{\mathcal{G}}$

Lemma 2 *Assume the joint probability distribution \mathbb{P} of the process X has density f with respect to Lebesgue measure on $\mathbb{R}^{p \times n}$. If \mathbb{P} factorizes according to two different subgraphs of \mathcal{G}_{full} , \mathcal{G}_1 and \mathcal{G}_2 , then \mathbb{P} factorizes according to $\mathcal{G}_1 \cap \mathcal{G}_2$.*

Lemma 3 (Conditional independence between non adjacent successive variables) *Let \mathcal{G} be a subgraph of \mathcal{G}_{full} according to which the probability distribution \mathbb{P} admits a BN representation. For any pair of successive variables (X_{t-1}^j, X_t^i) which are non adjacent in \mathcal{G} , we have*

$$X_t^i \perp\!\!\!\perp X_{t-1}^j \mid pa(X_t^i, \mathcal{G}) \quad \text{and} \quad X_t^i \perp\!\!\!\perp X_{t-1}^j \mid pa(X_t^i, \mathcal{G}) \cup S,$$

for all S subset of $\{X_u^k; k \in P, u < t\}$.

The proof of these two lemmas is in Appendix. For an illustration of Lemma 3, assume \mathbb{P} admits a BN representation according to the DAG of Figure 3.2. There is no edge between X_t^3 and X_{t+1}^1 in this DAG. Now consider in Figure 3.4 the moral graph of the smallest ancestral graph containing X_t^3 , X_{t+1}^1 and the parents (X_t^1, X_t^2) of X_{t+1}^1 . The set (X_t^1, X_t^2) blocks all paths between X_t^3 and X_{t+1}^1 . From Proposition 4, we have $X_{t+1}^1 \perp\!\!\!\perp X_t^3 \mid pa(X_{t+1}^1, \mathcal{G})$.

It follows directly from Lemma 2 that, among the DAGs included in \mathcal{G}_{full} , it exists a minimal DAG, denoted by $\tilde{\mathcal{G}}$, according to which the probability distribution \mathbb{P} factorizes. From Lemma 3, the set of edges of $\tilde{\mathcal{G}}$ is exactly the set of full order conditional dependencies given the past of the process as set up in the next proposition.

Let $P_j = P \setminus \{j\}$. We denote by $X_t^{P_j} = \{X_t^k; k \in P_j\}$ the set of $p - 1$ variables observed at time t .

Proposition 5 (BN representation according to $\tilde{\mathcal{G}}$, the smallest subgraph of \mathcal{G}_{full})

Whenever Assumptions 1 and 2 are satisfied, the probability distribution \mathbb{P} admits a BN representation according to DAG $\tilde{\mathcal{G}}$ whose edges describe exactly the full order conditional dependencies between successive variables X_{t-1}^j and X_t^i given the remaining variables $X_{t-1}^{P_j}$ observed at time $t - 1$,

$$\tilde{\mathcal{G}} = \left(X, \left\{ (X_{t-1}^j, X_t^i); X_t^i \not\perp\!\!\!\perp X_{t-1}^j \mid X_{t-1}^{P_j} \right\}_{i,j \in P, t \in N} \right),$$

Moreover, DAG $\tilde{\mathcal{G}}$ is the smallest subgraph of \mathcal{G}_{full} according to which \mathbb{P} admits a BN representation.

See Proof in Appendix. In DAG $\tilde{\mathcal{G}}$, the set of parents $pa(X_t^i, \tilde{\mathcal{G}})$ of each variable X_t^i is the smallest subset of X_{t-1} such that the conditional densities satisfy $f(X_t^i \mid pa(X_t^i, \tilde{\mathcal{G}})) = f(X_t^i \mid X_{t-1})$. The set of parents of each variable can be seen as the only variables on which this variable depends directly. So $\tilde{\mathcal{G}}$ is the DAG we want to infer to recover potential regulation relationships from gene expression time series. From Lemma 3, any pair of successive variables (X_{t-1}^j, X_t^i) which are non adjacent in $\tilde{\mathcal{G}}$ are conditionally independent given the parents of X_t^i ,

$$X_t^i \perp\!\!\!\perp X_{t-1}^j \mid pa(X_t^i, \tilde{\mathcal{G}}).$$

We will make use of this result in section 3.3 in order to define low order conditional independence DAGs for the inference of $\tilde{\mathcal{G}}$.

3.2.4 DAG $\tilde{\mathcal{G}}_{AR(1)}$ for an AR(1) process

Consider the following first order auto-regressive model,

AR(1) model

$$X_1 \sim \mathcal{N}(\mu_1, \Sigma_1) \tag{3.4}$$

$$\forall t > 1, \quad X_t = AX_{t-1} + B + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \Sigma), \tag{3.5}$$

$$\forall s, t \in N, \quad Cov(\varepsilon_t, \varepsilon_s) = \delta_{ts}\Sigma, \tag{3.6}$$

$$\forall s > t, \quad Cov(X_t, \varepsilon_s) = 0. \tag{3.7}$$

where $A = (a_{ij})_{1 \leq i \leq p, 1 \leq j \leq p}$ is a $p \times p$ matrix, $B = (b_i)_{1 \leq i \leq p}$ is a column vector of size p , $\Sigma = (\sigma_{ij})_{1 \leq i \leq p, 1 \leq j \leq p}$ is the error covariance matrix and for all s, t in N , $\delta_{ts} = \mathbf{1}_{\{s=t\}}$. Equation (3.7) implies that the coefficient matrices are uniquely determined from the covariance function of X_t .

This modeling assumes homogeneity across time (constant matrix A) and linearity of the dependency relationships. From (3.5) and (3.7), the model is first order Markovian and Assumption 1 is satisfied. From (3.6), Assumption 2 is satisfied whenever the error covariance matrix Σ is diagonal. Considering non correlated measurement errors between distinct genes is a strong assumption especially since microarray data contain several sources of noise including block effects. Nevertheless, assuming Σ diagonal is still reasonable after a normalization procedure.

From Proposition 5, the probability distribution of this AR(1) process factorizes according to a minimal DAG $\tilde{\mathcal{G}}_{AR(1)}$ whose edges correspond to the non-zero coefficients of matrix A . Indeed, if matrix Σ is diagonal, each element a_{ij} is the regression coefficient of the variable X_t^i on X_{t-1}^j given $X_{t-1}^{P_j}$, that is

$$a_{ij} = Cov(X_t^i, X_{t-1}^j | X_{t-1}^{P_j}) / Var(X_{t-1}^j | X_{t-1}^{P_j}).$$

So the set of null coefficients of the matrix A exactly describes the conditional independencies between successive variables,

$$\text{if } \Sigma \text{ is diagonal, we have } a_{ij} = 0 \Leftrightarrow \left\{ \forall t > 1, X_t^i \perp\!\!\!\perp X_{t-1}^j | X_{t-1}^{P_j} \right\}.$$

So DAG $\tilde{\mathcal{G}}_{AR(1)}$ has an edge between two successive variables X_{t-1}^j and X_t^i , for all $t > 1$, whenever the coefficient a_{ij} of the matrix A differs from zero,

$$\tilde{\mathcal{G}}_{AR(1)} := (X, \{(X_{t-1}^j, X_t^i) \text{ such that } a_{ij} \neq 0; t > 1, i, j \in P\}). \quad (3.8)$$

For an illustration, any AR(1) process whose matrix Σ is diagonal and matrix A has the following form,

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & 0 & 0 \\ 0 & a_{32} & 0 \end{pmatrix}$$

admits a BN representation according to the dynamic network of Figure 3.2 ($p = 3$).

3.3 Approximating $\tilde{\mathcal{G}}$ with DAGs $\mathcal{G}^{(q)}$

From Proposition 5, reverse discovering the DAG $\tilde{\mathcal{G}}$ requires to determine, for each variable X_t^i , the set of variables X_{t-1}^j observed at time $t-1$ which are conditionally dependent on X_t^i given the remaining variables $X_{t-1}^{P_j}$. Even under homogeneity assumption (see section 3.1), the available data of gene expression time series do not allow such testing. We still have to face the 'curse of dimension' as the number of genes p , is much higher than the number of measurements n . By extending the approach proposed by Wille et al. [WZV⁺04, WB06] to DBNs, we propose here an original approach for the inference of dynamic networks of high size by considering low order independencies.

3.3.1 Definition

We approximate DAG $\tilde{\mathcal{G}}$ of full order conditional dependence by the q^{th} order conditional dependence graph $\mathcal{G}^{(q)}$ (with $q < p$). In DAG $\mathcal{G}^{(q)}$, no edge is drawn between two successive variables X_{t-1}^j and X_t^i whenever it exists a subset X_{t-1}^Q of q variables among the $p-1$ variables $X_{t-1}^{P_j}$ such that X_{t-1}^j and X_t^i are conditionally independent given this subset. In short, DAGs $\mathcal{G}^{(q)}$ are defined as follows,

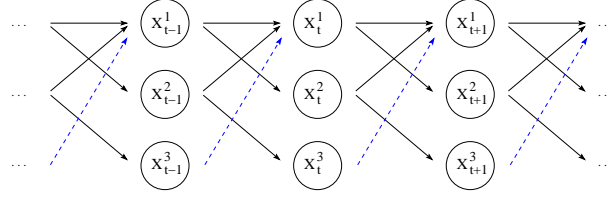


Figure 3.5: First-order conditional dependence DAG $\mathcal{G}^{(1)}$ (obtained from the DAG in Figure 3.2). The spurious dashed arrow may appear in $\mathcal{G}^{(1)}$.

Definition 6 q th-order conditional dependence DAG $\mathcal{G}^{(q)}$

$$\forall q < p, \quad \mathcal{G}^{(q)} = \left(X, \left\{ (X_{t-1}^j, X_t^i); \forall Q \subseteq P_j, |Q| = q, X_t^i \not\perp\!\!\!\perp X_{t-1}^j | X_{t-1}^Q \right\}_{i,j \in P, t \in N} \right).$$

Note that DAGs $\mathcal{G}^{(q)}$ offer a way of producing some dependence relationships between the variables but are not anymore associated with a BN representation which would call for more global relationships. However DAG $\tilde{\mathcal{G}}$, which allows a BN representation, corresponds to the $(p-1)^{th}$ order conditional dependence DAG $\mathcal{G}^{(p-1)}$.

In general, DAGs $\mathcal{G}^{(q)}$ differ from DAG $\tilde{\mathcal{G}}$. For instance, the approximation of the DAG of Figure 3.2 by the 1st order conditional dependence DAG may give birth to the spurious edge $X_{t-1}^3 \rightarrow X_t^1$, for all $t > 1$ (see Figure 3.5). Indeed, neither X_{t-1}^1 nor X_{t-1}^2 separates X_t^1 from X_{t-1}^3 in the smallest moral graph containing $X_t^1 \cup X_{t-1}^3 \cup X_{t-1}^1$ (resp. $X_t^1 \cup X_{t-1}^3 \cup X_{t-1}^2$). Nevertheless, if the vertices of $\tilde{\mathcal{G}}$ have few parents, DAGs $\mathcal{G}^{(q)}$ can bring relevant information on the topology of $\tilde{\mathcal{G}}$, even for small value of q . In the following, we give characterizations of low order conditional dependence DAGs $\mathcal{G}^{(q)}$ and analyze how good approximations they do offer.

3.3.2 A restricted number of parents

In the known gene regulation mechanisms, some genes regulate many other genes (e.g. single input modules in the transcriptional regulatory network of *S. Cerevisiae* [LRR⁺02]). Nevertheless, we do not expect a single gene to be regulated by a lot of genes at the same time. So the number of parents in gene interaction networks is expected to be relatively small. In this section, we analyze the properties of $\mathcal{G}^{(q)}$ when the number of parents in $\tilde{\mathcal{G}}$ is lower than q .

Let us denote by $N_{pa}(X_t^i, \tilde{\mathcal{G}})$ the number of parents of X_t^i in the DAG $\tilde{\mathcal{G}}$ and $N_{pa}^{Max}(\tilde{\mathcal{G}})$ the maximal number of parents of any variable X_t^i in $\tilde{\mathcal{G}}$,

$$N_{pa}(X_t^i, \tilde{\mathcal{G}}) = |pa(X_t^i, \tilde{\mathcal{G}})|, \quad N_{pa}^{Max}(\tilde{\mathcal{G}}) = \max_{i \in P, t \in N} (N_{pa}(X_t^i, \tilde{\mathcal{G}})).$$

The next results hold when the number of parents in $\tilde{\mathcal{G}}$ is restricted.

Proposition 6 If $N_{pa}(X_t^i, \tilde{\mathcal{G}}) \leq q$ then $\{(X_{t-1}^j, X_t^i) \notin E(\tilde{\mathcal{G}})\} \Rightarrow \{(X_{t-1}^j, X_t^i) \notin E(\mathcal{G}^q)\}$.

Corollary 1 For all $q \geq N_{pa}^{Max}(\tilde{\mathcal{G}})$, we have $\tilde{\mathcal{G}} \supseteq \mathcal{G}^{(q)}$.

Proposition 7 Let X be a Gaussian process. If $N_{pa}^{Max}(\tilde{\mathcal{G}}) \leq 1$ then $\tilde{\mathcal{G}} = \mathcal{G}^{(1)}$.

Consider a variable X_t^i having at most q parents in $\tilde{\mathcal{G}}$ ($q < p$). Let X_{t-1}^j be a variable observed at the previous time $t - 1$ and having no edge pointing towards X_t^i in $\tilde{\mathcal{G}}$. In the moral graph of the smallest ancestral set containing $X_t^i \cup X_{t-1}^j \cup pa(X_t^i, \tilde{\mathcal{G}})$, the set of parents $pa(X_t^i, \tilde{\mathcal{G}})$ separates X_t^i from X_{t-1}^j . From Proposition 4, we have $X_t^i \perp\!\!\!\perp X_{t-1}^j \mid pa(X_t^i, \tilde{\mathcal{G}})$. The number of parents $pa(X_t^i, \tilde{\mathcal{G}})$ is lower than q , so the edge $X_{t-1}^j \rightarrow X_t^i$ is not in $\mathcal{G}^{(q)}$. This establishes Proposition 6.

Consequently, if the maximal number of parents in $\tilde{\mathcal{G}}$ is lower than q then $\mathcal{G}^{(q)}$ is included in $\tilde{\mathcal{G}}$ (Corollary 1). In that case, $\mathcal{G}^{(q)}$ does not contain spurious edges.

The converse inclusion relationship is not true in general. Let $X_{t-1}^j \rightarrow X_t^i$ be an edge of $\tilde{\mathcal{G}}$, then X_t^i and X_{t-1}^j are conditionally dependent given the remaining variables $X_{t-1}^{P_j}$. It may however exist a subset of q variables X_{t-1}^Q , where Q is a subset of $P \setminus \{j\}$ of size q , such that X_t^i and X_{t-1}^j are conditionally independent with respect to this subset X_{t-1}^Q . Indeed, even though the topology of $\tilde{\mathcal{G}}$ allows to establish some conditional independencies, DAG $\tilde{\mathcal{G}}$ does not necessary allow to derive all of them. Two variables can be conditionally independent given a subset of variables whereas this subset does not separate these two variables in $\tilde{\mathcal{G}}$. Nevertheless, if each variable has at most *one* parent, the converse inclusion $\tilde{\mathcal{G}} \subseteq \mathcal{G}^{(1)}$ is true if the process is Gaussian and $q = 1$ (Proposition 7, see proof in Appendix). At a higher order, we need to assume that all conditional independencies can be derived from $\tilde{\mathcal{G}}$, that is \mathbb{P} is *faithful* to $\tilde{\mathcal{G}}$.

3.3.3 Faithfulness

Definition 7 (faithfulness, Spirtes [SGS93]) A distribution \mathbb{P} is **faithful** to a DAG \mathcal{G} if all and only the independence relationships true in \mathbb{P} are entailed by \mathcal{G} (as set up in Proposition 4).

Theorem 1 (Measure zero for unfaithful Gaussian (Spirtes [SGS93]) and discrete (Meek [Mee95]) distributions) Let $\pi_{\mathcal{G}}^N$ (resp. $\pi_{\mathcal{G}}^D$) be the set of linearly independent parameters needed to parameterize a multivariate normal distribution (resp. discrete distribution) \mathbb{P} which admits a factorization according to a DAG \mathcal{G} . The set of distributions which are unfaithful to \mathcal{G} is measure zero with respect to Lebesgue measure over $\pi_{\mathcal{G}}^N$ (resp. over $\pi_{\mathcal{G}}^D$).

If distribution \mathbb{P} is faithful to $\tilde{\mathcal{G}}$, then any subset $X_{t-1}^Q \subseteq X_{t-1}$, with respect to which X_t^i and X_{t-1}^j are conditionally independent, separates X_t^i and X_{t-1}^j in the moral graph of the smallest ancestral set containing $X_t^i \cup X_{t-1}^j \cup X_{t-1}^Q$. Under this assumption, we can derive interesting properties on $\tilde{\mathcal{G}}$ from the topology of low order dependence DAGs $\mathcal{G}^{(q)}$. As there is no way to assess a probability distribution to be faithful to a DAG, this assumption has often been criticized. Nevertheless, Theorem 1, established by Spirtes [SGS93] for Gaussian distribution and extended to discrete distribution by Meek [Mee95], makes this assumption reasonable at least in a measure-theoretic sense. Given that we consider a single distribution inherent to the studied process, the distribution \mathbb{P} is not necessary faithful to $\tilde{\mathcal{G}}$. Nevertheless, this assumption appears very reasonable and calls for careful interest. The next propositions are derived from faithfulness to $\tilde{\mathcal{G}}$.

Proposition 8 Assume \mathbb{P} is faithful to $\tilde{\mathcal{G}}$. For all $q < p$, we have $\tilde{\mathcal{G}} \subseteq \mathcal{G}^{(q)}$.

Proof. Let $(X_{t-1}^j, X_t^i) \in E(\tilde{\mathcal{G}})$. Assume that $(X_{t-1}^j, X_t^i) \notin E(\mathcal{G}^{(q)})$ then it exists a subset of q variables X_{t-1}^Q with respect to which X_{t-1}^j and X_t^i are conditionally independent. From faithfulness, the subset X_{t-1}^Q separates X_{t-1}^j and X_t^i in the moral graph of the smallest ancestral set containing $X_t^i \cup X_{t-1}^j \cup X_{t-1}^Q$. This contradicts the presence of the edge (X_{t-1}^j, X_t^i) in $\tilde{\mathcal{G}}$. ■

Corollary 2 Assume \mathbb{P} is faithful to $\tilde{\mathcal{G}}$. For all $q \geq N_{pa}^{Max}(\tilde{\mathcal{G}})$, we have $\tilde{\mathcal{G}} = \mathcal{G}^{(q)}$.

Proposition 9 Assume \mathbb{P} is faithful to $\tilde{\mathcal{G}}$. If $N_{pa}(X_t^i, \mathcal{G}^{(q)}) \leq q$ then $(X_{t-1}^j, X_t^i) \in E(\mathcal{G}^{(q)}) \Rightarrow (X_{t-1}^j, X_t^i) \in E(\tilde{\mathcal{G}})$.

Proof. From faithfulness, $\tilde{\mathcal{G}} \subseteq \mathcal{G}^{(q)}$. Then for all X_t^i , $N_{pa}(X_t^i, \tilde{\mathcal{G}}) \leq N_{pa}(X_t^i, \mathcal{G}^{(q)}) \leq q$. From Proposition 6, $(X_{t-1}^j, X_t^i) \notin E(\tilde{\mathcal{G}}) \Rightarrow (X_{t-1}^j, X_t^i) \notin E(\mathcal{G}^{(q)})$, that is $(X_{t-1}^j, X_t^i) \in E(\mathcal{G}^{(q)}) \Rightarrow (X_{t-1}^j, X_t^i) \in E(\tilde{\mathcal{G}})$.
■

Corollary 3 Assume \mathbb{P} is faithful to $\tilde{\mathcal{G}}$. For all $q \geq N_{pa}^{Max}(\mathcal{G}^{(q)})$, we have $\tilde{\mathcal{G}} = \mathcal{G}^{(q)}$.

Even though we expect the number of parents in a gene interaction networks to be upper bounded, the exact maximal number of parents $N_{pa}^{Max}(\tilde{\mathcal{G}})$ remains unknown. Nevertheless, if \mathbb{P} is faithful to $\tilde{\mathcal{G}}$, some edges of $\tilde{\mathcal{G}}$ can still be derived from the topology of q^{th} order conditional dependence DAGs $\mathcal{G}^{(q)}$ without knowing the maximal number of parents in $\tilde{\mathcal{G}}$. Indeed, from Proposition 9, the edges of DAG $\mathcal{G}^{(q)}$ pointing towards a variable having less than q parents in $\mathcal{G}^{(q)}$ are edges of $\tilde{\mathcal{G}}$ too.

3.4 Inferring $\tilde{\mathcal{G}}$

We introduced and characterized the q^{th} order dependence DAGs $\mathcal{G}^{(q)}$, for all $q < p$, in dynamic modeling. We now exploit our results to develop a non-Bayesian inference method for DAG $\tilde{\mathcal{G}}$. Let q_{max} be the maximal number of parents in $\tilde{\mathcal{G}}$. From Corollary 3, inferring $\tilde{\mathcal{G}}$ amounts to inferring $\mathcal{G}^{(q_{max})}$. However, the inference of $\mathcal{G}^{(q_{max})}$ requires to check, for each pair (i, j) , if there exists a subset $Q \subseteq P_j$ of dimension q_{max} such that $X_t^i \perp\!\!\!\perp X_{t-1}^j | X_{t-1}^Q$ for all $t > 1$. So, for each pair (i, j) , there are $\binom{q_{max}}{p-1}$ potential sets that can lead to conditional independence. To test each conditional independence given any possible subset of q_{max} variables is questionable both in terms of complexity and multiple testings.

To circumvent these issues, we propose to exploit the fact that the true model $\tilde{\mathcal{G}}$ is a subgraph of $\mathcal{G}^{(1)}$ (Proposition 8) to develop an inference procedure. Indeed, the inference of $\mathcal{G}^{(1)}$ is both the faster (complexity) and the most accurate (number of tests). So we set out a two step procedure: first to infer $\mathcal{G}^{(1)}$, second to infer $\tilde{\mathcal{G}}$ from the estimated DAG $\hat{\mathcal{G}}^{(1)}$. Nevertheless, DAG $\hat{\mathcal{G}}^{(1)}$ already offers a very good approximation of $\tilde{\mathcal{G}}$ when it is sparse (see Figure 3.7, left). We develop here the 2 step-procedure which is summarized in Figure 4.2.

3.4.1 Step 1: inferring $\mathcal{G}^{(1)}$

We evaluate the *likelihood* of an edge (X_{t-1}^j, X_t^i) by measuring the conditional dependence between the variables X_{t-1}^j and X_t^i given any variable X_{t-1}^k . Let $a_{ij|k}$ be the partial regression coefficient defined as follows,

$$X_t^i = m_{ijk} + a_{ij|k} X_{t-1}^j + a_{ik|j} X_{t-1}^k + \eta_t^{i,j,k},$$

where the rank of the matrix $(X_{t-1}^j, X_{t-1}^k)_{t \geq 2}$ equals 2 and the errors $\{\eta_t^{i,j,k}\}_{t \geq 2}$ are centered, have same variance and are not correlated.

We chose to measure the conditional dependence between the variables X_{t-1}^j and X_t^i given any variable X_{t-1}^k by testing null assumption $\mathcal{H}_0^{i,j,k}$: “ $a_{ij|k} = 0$ ”. To such an aim, we use one out of

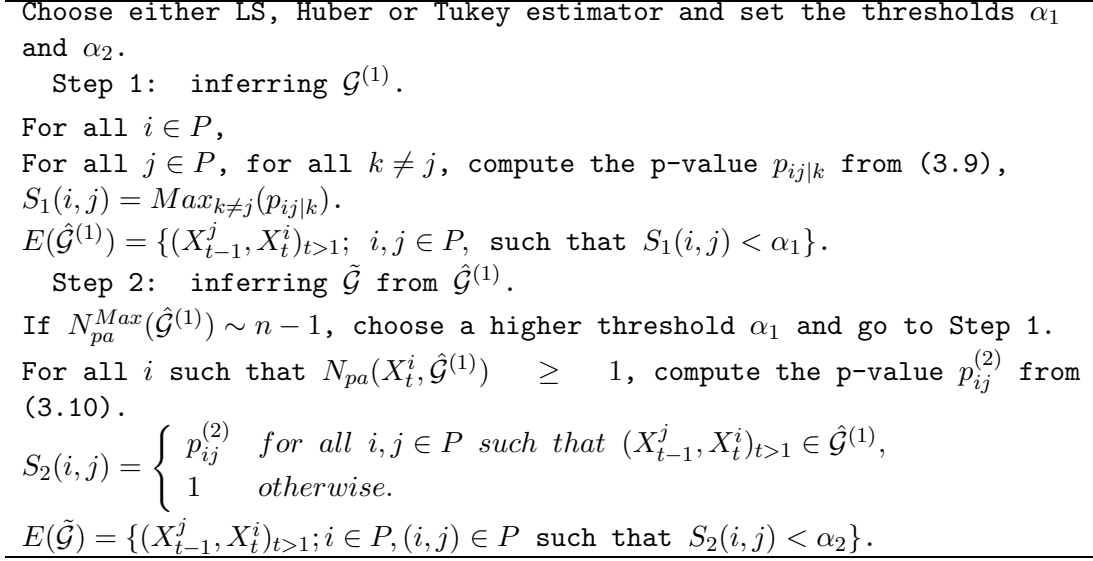


Figure 3.6: Algorithm

three M-estimators for this coefficient: either the familiar Least Square (LS) estimator, the *Huber* estimator, or the *Tukey bisquare* (or *biweight*) estimator. The two latter are robust estimators [Fox02]. Then for each $k \neq j$, we compute the estimates $\hat{a}_{ij|k}$ according to one of these three estimators and derive the p-value $p_{ij,k}$ from the standard significance test:

$$\text{under } (\mathcal{H}_0^{i,j,k}) : "a_{ij|k} = 0", \quad \frac{\hat{a}_{ij|k}}{\hat{\sigma}(\hat{a}_{ij|k})} \sim t(n-4), \quad (3.9)$$

where $t(n-4)$ refers to a student probability distribution with $n-4$ degrees of freedom and $\hat{\sigma}(\hat{a}_{ij|k})$ is the variance estimates for $\hat{a}_{ij|k}$.

Thus, we assign a score $S_1(i, j)$ to each potential edge (X_{t-1}^j, X_t^i) equal to the maximum $\text{Max}_{k \neq j}(p_{ij|k})$ of the $p-1$ computed p-values, that is the most favorable result to 1st order conditional independence. This procedure does not derive p-values for the edges but allows to order the possible edges of DAG $\mathcal{G}^{(1)}$ according to how likely they are. The smallest scores point out the most significant edges for $\mathcal{G}^{(1)}$. The inferred DAG $\hat{\mathcal{G}}^{(1)}$ contains the edges having a score below a chosen threshold α_1 . We compare the three estimators used for the inference of $\tilde{\mathcal{G}}$ in a simulation study in the next section (Figure 3.7, right).

3.4.2 Step 2: inferring $\tilde{\mathcal{G}}$

We use the inferred DAG $\hat{\mathcal{G}}^{(1)}$ as a reduction of the search space. Indeed, from faithfulness, $\tilde{\mathcal{G}} \subseteq \mathcal{G}^{(1)}$ (Proposition 8). Moreover, when DAG $\tilde{\mathcal{G}}$ is sparse, there are far fewer edges in $\mathcal{G}^{(1)}$ than in the complete DAG \mathcal{G}_{full} defined in subsection 3.2.2. Consequently, the number of parents of each variable in $\hat{\mathcal{G}}^{(1)}$ is much lower than n . Then model selection can be carried out by using standard estimation and tests among the edges of $\hat{\mathcal{G}}^{(1)}$. For each pair (i, j) such that the set of edges $(X_{t-1}^j, X_t^i)_{t>1}$ is in $\hat{\mathcal{G}}^{(1)}$, we denote $a_{ij}^{(2)}$ the following regression coefficient,

$$X_t^i = m_i + \sum_{j \in pa(X_t^i, \hat{\mathcal{G}}^{(1)})} a_{ij}^{(2)} X_{t-1}^j + \eta_t^i,$$

where the rank of the matrix $(X_{t-1}^j)_{t \geq 2, j \in pa(X_t^i, \hat{\mathcal{G}}^{(1)})}$ is $|pa(X_t^i, \hat{\mathcal{G}}^{(1)})|$ and the errors $\{\eta_t^i\}_{t \geq 2}$ are centered, have same variance and are not correlated. We assign to each edge of $\hat{\mathcal{G}}^{(1)}$ the score

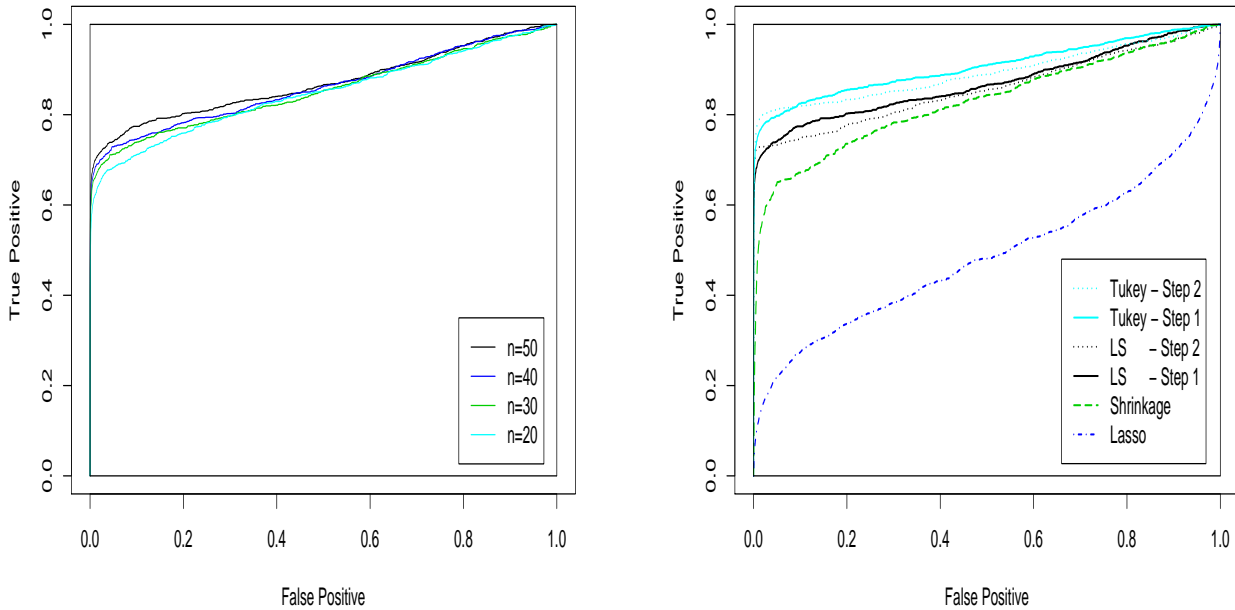


Figure 3.7: ROC curves for the inference of $\tilde{\mathcal{G}}$. Left: ROC curves obtained by $\mathcal{G}^{(1)}$ approximation (Step 1) with the Least Square estimator when $n = 20$ to 50 . Right: improvement obtained with both robust estimation and Step 2 of the procedure ($\alpha_1 = 0.9$, $n = 50$); comparison with Shrinkage and Lasso regression.

$S_2(i, j)$ equal to the p-value $p_{ij}^{(2)}$ derived from the significance test,

$$\text{under } (\mathcal{H}_0^{i,j}) : "a_{ij}^{(2)} = 0", \quad \frac{\hat{a}_{ij}^{(2)}}{\hat{\sigma}(\hat{a}_{ij}^{(2)})} \sim t(n-1 - |pa(X_t^i, \hat{\mathcal{G}}^{(1)})|). \quad (3.10)$$

The score $S_2(i, j) = 1$ is assigned to the edges that are not in $\hat{\mathcal{G}}^{(1)}$. The smallest scores point out the most significant edges. The inferred DAG for $\tilde{\mathcal{G}}$ contains the edges whose score is below a chosen threshold α_2 . We implemented this inference procedure and some analysis tools in the R package 'G1DBN'. It is distributed under the terms of the GNU General Public License and freely available from the R package archive (<http://cran.r-project.org>).

When $\tilde{\mathcal{G}}$ is sparse, Step 1 of the procedure gives a good estimation of $\tilde{\mathcal{G}}$ already (see ROC curves of Figure 3.7, left). Even better results can be obtained with the 2 step-procedure which requires to tune two parameters α_1 and α_2 . Parameter α_1 is the selection threshold of the edges of $\hat{\mathcal{G}}^{(1)}$ in step 1 (that is the dimension reduction threshold), whereas parameter α_2 is the selection threshold for the edges of $\tilde{\mathcal{G}}$ among the edges of DAG $\hat{\mathcal{G}}^{(1)}$. We study the impact of these parameters on the accuracy of the procedure through simulations in the next section.

3.5 Validation

3.5.1 Simulation study

We investigate the accuracy of the various approaches we proposed to recover DAG $\tilde{\mathcal{G}}$ for a multivariate AR(1) model. We randomly generate 100 sets of parameters $(A_{[p \times p]}, B, \Sigma)$ for $p = 50$. The gene regulation networks are known to be sparse. In accordance with this biological knowledge, each matrix A contains 2 % of non zero coefficients (sampled from uniform distribution). While keeping the number of parents low, this does not prevent it to be higher than one. Non zero coefficients were generated as follows, $a_{ij} \sim \mathcal{U}([-1.5; -0.5] \cup [0.5; 1.5])$, and we drew the mean $b_i \sim \mathcal{U}(0, 1)$ and error variance $\sigma_i \sim \mathcal{U}[0.03, 0.08]$ from uniform distributions. Time series were simulated under the corresponding multivariate AR(1) models for $n = 20$ to 50.

The left panel of Figure 3.7 displays the average ROC curves for the inference of DAG $\tilde{\mathcal{G}}$ obtained by $\mathcal{G}^{(1)}$ approximation (Step 1) with the LS estimator for $n = 20$ to 50. We ordered the edges (i, j) according to increasing maximal p-values $Max_{k \neq j}(p_{ij|k})$ for the significance tests of the partial regression coefficient estimates (see section 3.4.1 for details). For a very low false positive (FP) rate, the true positive (TP) rate rises 70 % for the longer time series. Even when $n = 20$, which is about the maximal length of the available time series gene expression data, the TP rate reaches almost 60 % whereas the FP rate remains almost null.

These results can still be improved. As an illustration, the right panel of Figure 3.7 displays average ROC curves obtained after the first or the second step of the procedure with either LS or Tukey bisquare estimates when $n = 50$. The ROC curves obtained with Huber estimates are very close to the Tukey bisquare curves and do not appear on this graph for sake of clarity. The solid black curve is the ROC curve obtained after step 1 computed with the LS estimator. Still with Step 1 only, the Tukey estimator allows to obtain better results (see the solid light line). In both cases, Step 2 (dotted lines) still higher the ROC curves for the first selected edges. So both Step 2 and robust estimation allows to higher the ROC curves, at least while keeping the FP rate very small. A similar improvement is obtained for others values of n .

We now recall the definitions:

$$TP = \frac{\text{Nb of true positive edges}}{\text{Nb of edges in the model}}, \quad PPV = \frac{\text{Nb of true positive edges}}{\text{Nb of selected edges}},$$

and examine more precisely the interest of Step 2. We notably analyze the impact of both thresholds α_1 and α_2 . The first step (inference of $\mathcal{G}^{(1)}$) allows to obtain a good TP/FP ratio already. Nevertheless the Positive Predictive Value (PPV) deteriorates very quickly when the threshold α_1 increases. As an illustration, see the dotted lines in the left panel of Figure 3.8 which shows the value of both the true positive rate and the PPV after Step 1 according to the threshold α_1 . After Step 1 only, the PPV is high for small values of α_1 , but then only a rather small percentage of edges is detected: for $\alpha_1 = 0.05$, $PPV = 90\%$ and $TP = 60\%$. On the contrary, when α_1 is high, even though the PPV is very small, the TP rate can reach very high values (up to 83 % for $\alpha_1 = 0.9$). Here comes the interest of the second step of the procedure. Indeed, even for high value of α_1 , the number of edges in $\hat{\mathcal{G}}^{(1)}$ - and consequently the number of parents - is much lower than the initial number of potential edges (\mathcal{G}_{full}). The dimension is then reduced in proportion and a second selection step can be carried out by using standard significance tests (see section 3.4.2). As appearing in black solid lines on Figure 3.8 (left), Step 2 allows to increase the PPV while the TP rate stays high. For $\alpha_1 = 0.85$ and $\alpha_2 = 0.01$, the true positive rate and the PPV both reach 75%. Both thresholds have to be tuned according to the objective.

Note that the procedure performs well even when there are several parents in the true DAG $\tilde{\mathcal{G}}$. The right panel of Figure 3.8 shows the positive predictive value (PPV) for the inferred DAG

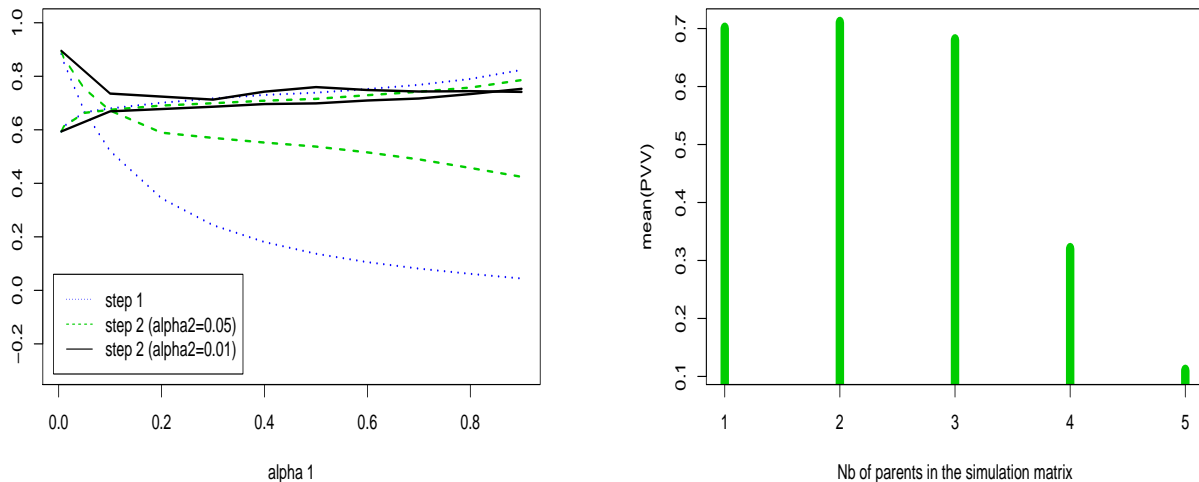


Figure 3.8: Left: TP rate (increasing curves) and PPV (decreasing curves) according to the threshold α_1 (n=50). Right: PPV according to the number of parents in the simulation DAG in the particular case $\alpha_1 = 0.9$, $\alpha_2 = 0.05$ (n=50).

$\tilde{\mathcal{G}}$ according to the number of parents in the simulation model for $\alpha_1 = 0.9$ and $\alpha_2 = 0.05$. Up to 3 parents, the PPVs are comparable and reach 70% on average.

Finally, we compare our approach with two reference methods for model selection in multivariate AR(1) process: the shrinkage approach by Opgen-Rhein and Strimmer and the Lasso regression.

Opgen-Rhein and Strimmer [ORS07] recently proposed a model selection procedure based on an analytic shrinkage approach. The procedure first consists in computing the partial correlation coefficients from the shrinkage estimates of the partial regression coefficients, and second in selecting the edges with a *local* false discovery rate approach [Efr05]. We carried out model selection in the simulated data with the R package they implemented. The ROC curve obtained by this shrinkage approach appears in dashed line (- -) in the right panel of figure 3.7.

The L1 regression (Lasso) [Tib96] combines shrinkage and model selection. This approach offers the advantage that it automatically sets many regression coefficients to zero. We carried out Lasso regression with the LARS package [EHJT04]. We chose the penalty by cross-validation. As proposed by Opgen-Rhein and Strimmer, we computed partial correlation coefficients from the Lasso estimates and drew ROC curves by ordering the edges according to the absolute value of the corresponding partial correlation coefficient. The ROC curve for the Lasso approach appears in dashed-dotted line (-.-) in the right panel of figure 3.7.

Our procedure outperforms these two approaches. When using the 2 step-procedure with robust estimation, we reach 80% TP whereas the FP rate is almost null. The accuracy of our procedure comes from the increase of precision thanks to the dimension reduction. Indeed, this selection approach is based on 1st order conditional independence consideration. This allows to carry out significance testing in a model of dimension 4 (see section 3.4.1). This represents a drastic dimension reduction and makes the testing much more powerful. Indeed, even if there are more edges in $\mathcal{G}^{(1)}$ than in the true DAG $\tilde{\mathcal{G}}$ (Proposition 8), Step 1 of the procedure is very sensitive already (see Figure 3.7).

Even though the Shrinkage approach improves a lot the precision of the estimation of each par-

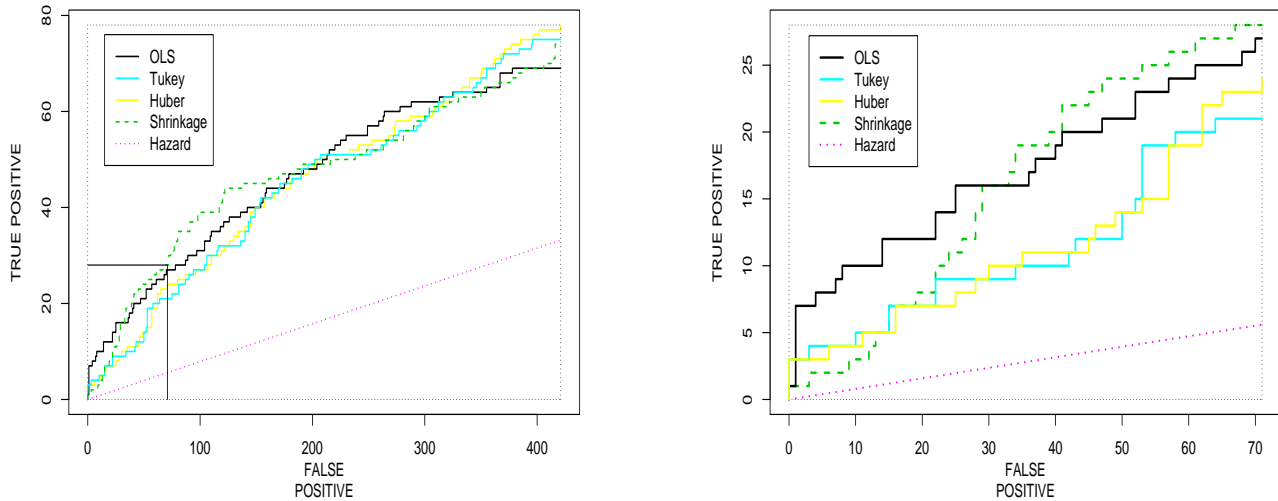


Figure 3.9: Four ROC curves respectively obtained with LS, Tukey, Huber based 1st order partial dependence inference procedure and Opgen-Rhein and Strimmer’s shrinkage approach applied to Spellman’s yeast cell cycle data. Left: the 500 first selected edges. Right: Focus on the 100 first selected edges.

tial correlation coefficient in comparison with standard methods, to consider 1st order conditional independence seems to be more powerful for the edge detection.

As for the Lasso, one major drawback lies in the fact that the edge selection is done vertex by vertex whereas the DAG $\tilde{\mathcal{G}}$ is globally but not uniformly sparse. As a consequence, the Lasso tends to uniformly reduce the number of parents of each vertex instead of only keeping small the total number of edges.

3.5.2 Analysis of microarray time course data sets

Spellman’s Yeast cell cycle data set

We apply the proposed method to the *Saccharomyces cerevisiae* cell cycle data collected by Spellman et al. [SSZ⁺98]. In the α Factor-based synchronization data (18 time points), we focus on the data set containing the 792 genes that demonstrated consistent periodic changes in transcription level. We only allow the 9 identified Transcription Factors (ACE2, FKH1, FKH2, MBP1, MCM1, NDD1, SWI4, SWI5, SWI6) that have been identified to have roles in regulating transcription of yeast genes [SBH⁺01] to be the possible regulators and try to infer their targets. We set the threshold α_1 to 0.05 for the inference of $\mathcal{G}^{(1)}$. From this DAG, we compute the p-value $p_{ij}^{(2)}$ to infer the true DAG $\tilde{\mathcal{G}}$. We compare the results of our 1st order partial independence inference procedure for the three estimates LS, Tukey and Huber as exposed in the previous section. We also inferred a DAG with the shrinkage approach by Opgen-Rhein and Strimmer [ORS07]. Validation is obtained from both the Yeastract database [TM06] and the targets of the cell cycle activators identified by Simon et al. [SBH⁺01].

Figure 3.9 displays the four obtained ROC curves. The left panel displays ROC curves for the 500 first selected edges, the right panel focus on the 100 first selected ones. Surprisingly, when using our 1st order partial dependence approach, the LS estimates outperforms robust estimates (according to Yeastract validation). However, the shrinkage approach and the LS-based 1st order

Table 3.2: First selected edges ($\alpha_1 = 0.05$) and validation (1/0=True/False regulation relationship).

TF	Target	$p_{ij}^{(2)}$	Validation
FKH2	KIP2	6.53e-06	1
SWI4	SVS1	6.87e-06	1
SWI4	AXL2	2.81e-05	0
SWI4	RKM1	3.85e-05	0
FKH2	CDC5	5.13e-05	1
FKH2	OGG1	5.21e-05	0
SWI4	SMC3	7.96e-05	0
FKH2	CLB2	9.10e-05	1
FKH2	SRC1	9.73e-05	1
SWI4	MSH2	1.16e-04	0

Table 3.3: Results of the inference method applied to the 792 genes of Spellman’s yeast cell cycle data ($\alpha_1 = 0.05$). Tuning α_2 allows to choose between TP rate and PPV.

α_2	TP edges	PPV
10^{-3}	25	40 %
10^{-2}	47	30 %
10^{-1}	60	18 %

partial dependence approach led to comparable results: the 1st order partial dependence approach performs better for the 50 first selected edges and the shrinkage approach performs better for the 50 next selected edges.

We now detail the results obtained with our LS-based 1st order partial dependence approach. The first selected edges appear in Table 3.2. The 4th column indicates whether or not a selected edge is a known regulation relationship. For $\alpha_2 = 0.001$, this procedure allows to detect 25 known regulation relationships (TP edges) with a PPV of 40%. The results for different values of α_2 appear in Table 3.3. When increasing α_2 , more edges are detected while the specificity stays acceptable comparative with other studies. Indeed, in one of the last in date DBN inference approach applied to the yeast cell cycle, Zou and Conzen [ZC05] reduced their analysis to a subset of only 116 regulated genes and compare their approach with Murphy’s Bayesian Network Toolbox [MM99]. By specifying the 9 identified TFs, Zou and Conzen correctly identify 46 edges with a PPV of 40% with their own procedure whereas they only obtain 18 correct edges with a PPV of 11% with Murphy’s BNT.

Diurnal cycle on the starch metabolism of *Arabidopsis Thaliana*

We applied our inference procedure to expression time series data generated by Smith et al. [SFC⁺04] to investigate the impact of the diurnal cycle on the starch metabolism of *Arabidopsis Thaliana*. We restricted our study to the 800 genes selected by Opgen-Rhein and Strimmer [ORS07] as having periodic expression profiles. The data are available in the GeneNet R package at <http://strimmerlab.org/software/genenet/html/arh800.html> or in a longitudinal format in our R package G1DBN (arth800line).

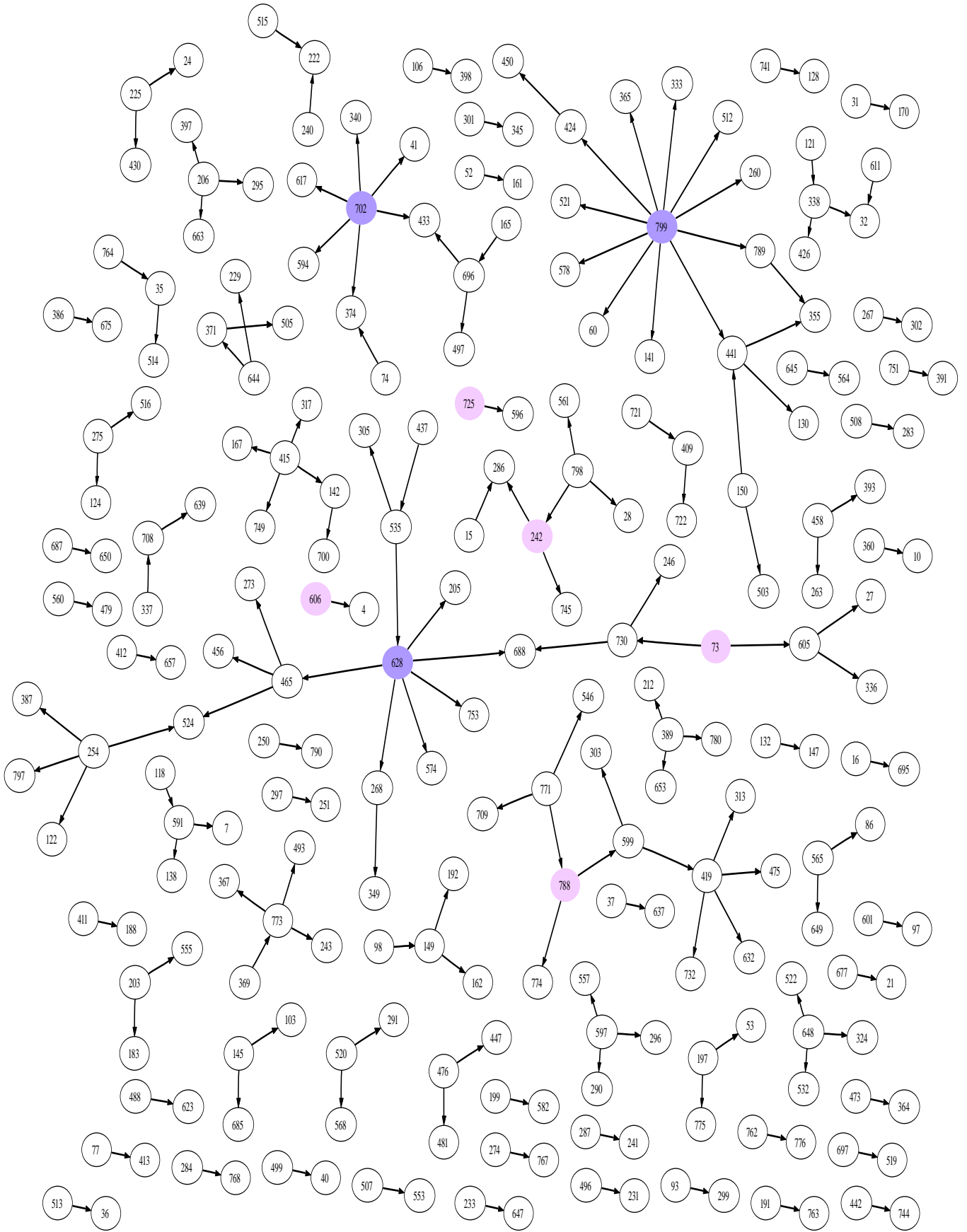


Figure 3.10: Inferred DAG $\tilde{\mathcal{G}}$ for $\alpha_1 = 0.1, \alpha_2 = 0.001$ (168 edges).

We choose a high 1st Step threshold $\alpha_1 = 0.1$ in order to maximize the chance that $\mathcal{G}^{(1)}$ contains the true DAG $\tilde{\mathcal{G}}$. For a 2nd Step threshold $\alpha_2 = 0.001$, we obtain the DAG $\tilde{\mathcal{G}}$ which appears in Figure 3.10. This DAG contains 168 edges implicating 236 different genes. 100 nodes are parent, 159 are child, 23 are both parent and child. The network differs from the network inferred by Opgen-Rhein and Strimmer [ORS07] but we still recover a network with a “hub” connectivity structure.

Among the ‘parent’ nodes in the network $\tilde{\mathcal{G}}$, the two proteins having the most ‘children’ (node 799 and 628) are known to be implicated in the starch metabolism. Indeed, node 799, which has 11 ‘children’ in $\tilde{\mathcal{G}}$, refers to DPE2 (DISPROPORTIONATING ENZYME 2), an essential component of the pathway from starch to sucrose and cellular metabolism in leaves at night. Then node 628 (6 children in $\tilde{\mathcal{G}}$) is a transferase (At5g24300) implicated in the starch synthase. Node 702 which is an unknown protein (At5g58220) has also 6 children in $\tilde{\mathcal{G}}$. These three nodes are dark-colored in the DAG of figure 3.10. The remaining ‘parent’ nodes have from 1 to 4 ‘children’. Among them, two are already identified as TFs and three as DNA binding proteins (see Table 3.4). These five nodes are light-colored in the DAG of figure 3.10. Finally a list of 28 unknown proteins have been selected as parents in the inferred DAG $\tilde{\mathcal{G}}$.

Complete results appear in the supplementary information available at <http://stat.genopole.cnrs.fr/~slebre/arth800DBN.pdf>. This notably displays the complete list of the unknown proteins selected as parents in the inferred DAG (section 2), the list of the parent nodes according to their number of target nodes (section 3) and the list of the edges ordered by decreasing significance (section 4) and by increasing past node number (section 5). The description of the 800 genes can be obtained from the GeneNet R package or at <http://stat.genopole.cnrs.fr/~slebre/arth800descr.pdf>.

Conclusion

In this paper, we first introduce a DBN modeling of gene expression time series which offers straight interpretation in terms of conditional dependence between gene expression levels. Then we define and characterize low order conditional dependence DAGs for dynamic networks. They offer a very good approximation of sparse DAGs.

From these results, we develop a novel inference method for dynamic genetic networks which makes it possible to face with the ‘small n , large p ’ estimation case. Our procedure proved to be powerful on both simulated and real data analysis. This approach based on the consideration of low order conditional dependence notably outperforms model selection based on shrinkage or lasso estimates.

We point out that robust estimators appeared very efficient for the detection of the edges. An interesting direction for further research lies in investigating which measures of the dependence in gene expression data are the more pertinent.

Acknowledgments

I would like to thank Catherine Matias and Bernard Prum for many stimulating and constructive discussions on this work.

Table 3.4: List of the 5 proteins selected as parents which have been identified as Transcription Factor or DNA binding.

Node	Gene Name	Description
73	AT2G43010-TAIR-G	PIF4 (PHYTOCHROME INTERACTING FACTOR 4); DNA binding / transcription factor; Isolated as a semidominant mutation defective in red - light responses. Encodes a nuclear localized bHLH protein that interacts with active PhyB protein. Negatively regulates phyB mediated red light responses.
242	AT1G05900-TAIR-G	DNA binding / endonuclease; endonuclease-related, similar to endonuclease III (Homo sapiens) GI:1753174; contains Pfam profile PF00633: Helix-hairpin-helix motif
606	AT5G10400-TAIR-G	DNA binding; histone H3, identical to several histone H3 proteins, including Zea mays SP—P05203, Medicago sativa GI:166384, Encephalartos altensteinii SP—P08903, Pisum sativum SP—P02300; contains Pfam profile PF00125 Core histone H2A/H2B/H3/H4
725	AT5G65360-TAIR-G	DNA binding; histone H3, identical to histone H3 from Zea mays SP—P05203, Medicago sativa GI:166384, Encephalartos altensteinii SP—P08903, Pisum sativum SP—P02300; contains Pfam profile PF00125 Core histone H2A/H2B/H3/H4
788	At4g14410-MinT-G	putative bHLH transcription factor (bHLH104)

3.6 Appendix: some additional proofs.

Proof of Lemma 2. Consider a discrete-time stochastic process $X = \{X_t^i; i \in P, t \in N\}$ whose joint probability \mathbb{P} distribution has the density f with respect to Lebesgue measure on $\mathbb{R}^{p \times n}$.

Let \mathcal{G}_1 and \mathcal{G}_2 be two different subgraphs of \mathcal{G}_{full} according to which the joint probability distribution \mathbb{P} factorizes. Let i in P , t in N , we consider the random variable X_t^i .

We denote as follows,

- the following subsets of P ,

$$pa_1 = \{j \in P; X_{t-1}^j \in pa(X_t^i, \mathcal{G}_1)\}$$

$$\overline{pa}_1 = P \setminus \{pa_1\}$$

$$pa_2 = \{j \in P; X_{t-1}^j \in pa(X_t^i, \mathcal{G}_2)\}$$

$$\overline{pa}_2 = P \setminus \{pa_2\}$$

- and the densities of the joint or marginal probability distributions of (X_t^i, X_{t-1}) ,
 $g : \mathbb{R}^{p+1} \rightarrow \mathbb{R}$ the density of the joint probability distribution of (X_t^i, X_{t-1}) ,
 g^i the density of the probability distribution of X_t^i ,
 g^P the density of the joint probability distribution of (X_{t-1}) ,
 g^{i,pa_1} the density of the joint probability distribution of $(X_t^i, X_{t-1}^{pa_1}) = (X_t^i, pa(X_t^i, \mathcal{G}_1))$,
 $g^{i,\overline{pa_2}}$ the density of the joint probability distribution of $(X_t^i, X_{t-1}^{\overline{pa_2}}) = (X_t^i, X_{t-1} \setminus \{pa(X_t^i, \mathcal{G}_2)\})$,
etc...

In the following, $y \in \mathbb{R}$, $x = (x_1, \dots, x_p) \in \mathbb{R}^p$ and we denote by $x_{pa_1} = \{x_j; j \in pa_1\} \in \mathbb{R}^{|pa_1|}$ (Thus $x = (x_{pa_1}, x_{\overline{pa_1}}) = (x_{pa_2}, x_{\overline{pa_2}}) \in \mathbb{R}^p$). As the probability distribution \mathbb{P} factorizes according to \mathcal{G}_1 , we derive from the DAG theory the conditional independence,

$$X_t^i \perp\!\!\!\perp X_{t-1}^{\overline{pa_1}} | X_{t-1}^{pa_1},$$

that is,

$$\forall y \in \mathbb{R}, \forall x \in \mathbb{R}^p, \quad \frac{g(y, x)}{g^P(x)} = \frac{g^{i,pa_1}(y, x_{pa_1})}{g^{pa_1}(x_{pa_1})}.$$

Equivalent results can be derived from the factorization according to \mathcal{G}_2 giving,

$$\forall y \in \mathbb{R}, x \in \mathbb{R}^p, N \quad g^{i,pa_2}(y, x_{pa_2}) = \frac{g^{i,pa_1}(y, x_{pa_1})}{g^{pa_1}(x_{pa_1})} g^{pa_2}(x_{pa_2}).$$

By taking the integral with respect to $x_{pa_2 \cap \overline{pa_1}}$, we write for all $y \in \mathbb{R}$, for all $x_{pa_1 \cup pa_2} \in \mathbb{R}^{|pa_1 \cup pa_2|}$,

$$\begin{aligned} \int g^{i,pa_2}(y, x_{pa_2}) d(x_{pa_2 \cap \overline{pa_1}}) &= \int \frac{g^{i,pa_1}(y, x_{pa_1})}{g^{pa_1}(x_{pa_1})} g^{pa_2}(x_{pa_2}) d(x_{pa_2 \cap \overline{pa_1}}) \\ g^{i,pa_1 \cap pa_2}(y, x_{pa_1 \cap pa_2}) &= \frac{g^{i,pa_1}(y, x_{pa_1})}{g^{pa_1}(x_{pa_1})} g^{pa_1 \cap pa_2}(x_{pa_1 \cap pa_2}) \end{aligned}$$

Finally we have,

$$\forall y \in \mathbb{R}, \forall x \in \mathbb{R}^p, \quad \frac{g(y, x)}{g^P(x)} = \frac{g^{i,pa_1 \cap pa_2}(y, x_{pa_1 \cap pa_2})}{g^{pa_1 \cap pa_2}(x_{pa_1 \cap pa_2})},$$

that is the conditional density of the probability distribution of X_t^i given X_{t-1} is the conditional density of the probability distribution of X_t^i given $X_{t-1}^{pa_1 \cap pa_2}$. Then \mathbb{P} factorizes according to $\mathcal{G}_1 \cap \mathcal{G}_2$.
■

Proof of Lemma 3. Assume \mathbb{P} admits a BN representation according to \mathcal{G} , a subgraph of \mathcal{G}_{full} . Let X_{t-1}^j and X_t^i be two *non adjacent* vertices of \mathcal{G} (there is no edge between them in \mathcal{G}) and consider the moral graph $(\mathcal{G}_{An}(X_t^i \cup X_{t-1}^j \cup pa(X_t^i, \mathcal{G})))^m$ of the smallest ancestral set containing the variables X_t^i , X_{t-1}^j and the parents $pa(X_t^i, \mathcal{G})$ of X_t^i in \mathcal{G} . As DAG \mathcal{G} is a subgraph of \mathcal{G}_{full} , the set of parents $pa(X_t^i, \mathcal{G})$ blocks all paths between X_{t-1}^j and X_t^i in the moral graph $(\mathcal{G}_{An}(X_t^i \cup X_{t-1}^j \cup pa(X_t^i, \mathcal{G})))^m$. From Proposition 4, this establishes the conditional independence $X_t^i \perp\!\!\!\perp X_{t-1}^j | pa(X_t^i, \mathcal{G})$.

This result holds for the conditioning according to any subset $S \subseteq \{X_u^k; k \in P, u < t\}$. ■

Proof of Proposition 5.

First, we show that \mathbb{P} admits a BN representation according to $\tilde{\mathcal{G}}$. Let $i, j \in P$ such that $X_t^i \perp\!\!\!\perp X_{t-1}^j | X_{t-1}^{P_j}$, then we have,

$$f(X_t^i | X_{t-1}) = f(X_t^i | X_{t-1}^{P_j}).$$

Under Assumptions 1 and 2, from Lemma 1 and Proposition 3, \mathbb{P} admits a BN representation according to the DAG $(X, E(\mathcal{G}_{full}) \setminus (X_{t-1}^j, X_t^i))$ which has the edges of \mathcal{G}_{full} except for the edge (X_{t-1}^j, X_t^i) . This holds for any pair of successive variables that are conditionally independent. Consequently, from Lemma 2, \mathbb{P} admits a BN representation according to the intersection of the DAG $(X, E(\mathcal{G}_{full}) \setminus (X_{t-1}^j, X_t^i))$ for any pair (X_t^i, X_{t-1}^j) such that $X_t^i \perp\!\!\!\perp X_{t-1}^j | X_{t-1}^{P_j}$, that is DAG $\tilde{\mathcal{G}}$.

Second, DAG $\tilde{\mathcal{G}}$ cannot be reduced. Indeed, let (X_{t-1}^l, X_t^k) be an edge of $\tilde{\mathcal{G}}$ and assume that \mathbb{P} admits a BN representation according to $\tilde{\mathcal{G}} \setminus (X_{t-1}^l, X_t^k)$, that is $\tilde{\mathcal{G}}$ reduced from the edge (X_{t-1}^l, X_t^k) . From Lemma 3, we have $X_t^k \perp\!\!\!\perp X_{t-1}^l | X_{t-1}^{P_l}$, which contradicts $(X_{t-1}^l, X_t^k) \in V(\tilde{\mathcal{G}})$ (i.e. $X_t^k \not\perp\!\!\!\perp X_{t-1}^l | X_{t-1}^{P_l}$).

■

Proof of Proposition 7.

First, from Corollary 1, $\tilde{\mathcal{G}} \supseteq \mathcal{G}^{(1)}$.

Second, let X be a Gaussian process and $(X_{t-1}^j, X_t^i) \in E(\tilde{\mathcal{G}})$, then according to Proposition 5, $X_t^i \not\perp\!\!\!\perp X_{t-1}^j | X_{t-1}^{P_j}$. Since X is Gaussian, this implies $Cov(X_t^i, X_{t-1}^j | X_{t-1}^{P_j}) \neq 0$.

Now assume that it exists $k \neq j$, such that $X_t^i \perp\!\!\!\perp X_{t-1}^j | X_{t-1}^k$ ie $(X_{t-1}^j, X_t^i) \notin E(\mathcal{G}^{(1)})$. We are going to prove that this contradicts $Cov(X_t^i, X_{t-1}^j | X_{t-1}^{P_j}) \neq 0$. Let l be an element of $P \setminus \{j, k\}$. The conditional covariance $Cov(ij|k, l) = Cov(X_t^i, X_{t-1}^j | X_{t-1}^k, X_{t-1}^l)$ writes,

$$\begin{aligned} Cov(ij|k, l) &= Cov(X_t^i, X_{t-1}^j | X_{t-1}^k) - \frac{Cov(X_t^i, X_{t-1}^j | X_{t-1}^k) Cov(X_{t-1}^j, X_{t-1}^l | X_{t-1}^k)}{Var(X_{t-1}^l | X_{t-1}^k)}, \\ &= Cov(X_t^i, X_{t-1}^j | X_{t-1}^k) \times \left[1 - \frac{(Cov(X_{t-1}^j, X_{t-1}^l | X_{t-1}^k))^2}{Var(X_{t-1}^j | X_{t-1}^k) Var(X_{t-1}^l | X_{t-1}^k)} \right] \\ &\quad - \frac{Cov(X_{t-1}^j, X_{t-1}^l | X_{t-1}^k) Cov(X_t^i, X_{t-1}^l | X_{t-1}^k, X_{t-1}^j)}{Var(X_{t-1}^l | X_{t-1}^k)}. \end{aligned}$$

However both terms in the latter expression of $Cov(ij|k, l)$ are null:

- since $X_t^i \perp\!\!\!\perp X_{t-1}^j | X_{t-1}^k$, then $Cov(X_t^i, X_{t-1}^j | X_{t-1}^k) = 0$,
- as $N_{pa}^{Max}(\tilde{\mathcal{G}}) \leq 1$, X_{t-1}^j is the only parent of X_t^i in $\tilde{\mathcal{G}}$. So the variable X_{t-1}^j and thus also the set (X_{t-1}^j, X_{t-1}^k) blocks all paths between X_{t-1}^j and X_t^i in the moral graph of the smallest ancestral set containing $X_t^i \cup X_{t-1}^{j,k,l}$. Then we have, $X_t^i \perp\!\!\!\perp X_{t-1}^l | \{X_{t-1}^j, X_{t-1}^k\}$, that is $Cov(X_t^i, X_{t-1}^l | X_{t-1}^k, X_{t-1}^j) = 0$.

Then $Cov(ij|k, l) = 0$. By induction, we obtain $Cov(X_t^i, X_{t-1}^j | X_{t-1}^{P_j}) = 0$ leading to a contradiction with $(X_{t-1}^j, X_t^i) \in E(\tilde{\mathcal{G}})$. Therefore $(X_{t-1}^j, X_t^i) \in \mathcal{G}^{(1)}$ and $\tilde{\mathcal{G}} \subseteq \mathcal{G}^{(1)}$.

■

Chapter 4

Inferring time-dependent networks from Systems Biology Time Course Data with reversible jump MCMC.

This chapter introduces a work motivating the very soon submission of an article written in collaboration with Gaëlle Lelandais.

4.1 Introduction

4.1.1 Background

Most of the time series data analysis aim at observing a chronological process, that is a process that can be divided into several phases. The yeast cell-cycle is an excellent illustration of this phenomenon. Indeed, the available time series [SSZ⁺98] cover the whole cycle which can at least be divided into 4 phases (G1, S, G2, M). Consequently the effective regulation relationships may change along a single process and so do the topology of the network representing these relationships across time.

The phases of the yeast cell-cycle are well known, but this is not the case of most of the biological processes under study. Can we manage to detect phases if they are a priori unknown? Up to now, many procedures have been proposed to infer dynamic networks from time series data [KIM03, BFG⁺05, ORS07, Leb07], but the underlying network is assumed to be homogeneous across time. We propose here to infer a time-dependent network from temporal data, that is a model which allows the relationships between genes to vary across the whole process. Of course, the approach requires more data than to infer homogeneous network. We need either repeated time point measurements or more time points during the observed process.

Very recently, Rao et al. [RHISE07] also proposed to infer such a time varying network from gene expression data. To such an aim, they first detect changepoints position for each gene, then they cluster genes having similar behavior and sharing the same changepoints and finally they infer the network topology describing relationships between genes within each cluster. We propose here to *simultaneously* infer the changepoint position and the structure of the networks within phases thanks to a reversible jump Monte Carlo Markov Chain (MCMC) approach. Indeed, we face a joint detection/estimation problem where the dimension is typically not fixed. Reversible jump MCMC methods were especially introduced by Green [Gre95] to jointly solve these issues. Thus we develop an efficient stochastic algorithm based on two embedded reversible jump MCMC

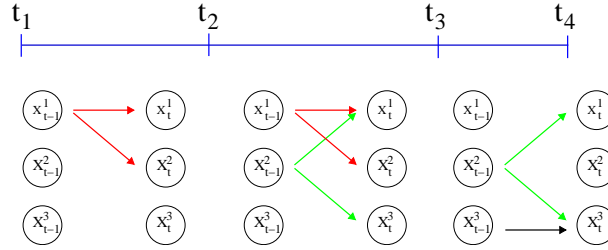


Figure 4.1: Non homogeneous network with 2 changepoints.

levels. One level is for multiple changepoint detection and the other one for model selection within the phases delimited by the changepoints.

This approach is used to analyze the response developed by the yeast *S. cerevisiae* after addition of the antimetabolic drug benomyl from multi-knockout data. The first results notably allowed to point out the chronological effect of transcription factor YAP1 deletion.

4.1.2 Non homogeneous network Model

Let p be the number of 'target' genes, q the number of 'factor' genes. We denote by $P = \{1, \dots, p\}$ and $Q = \{1, \dots, q\}$ the set of target and factor genes respectively. We consider two random variables vectors $Y_t = \{Y_t^i; i \in P\}$ and $X_t = \{X_t^j; j \in Q\}$ referring to an expression measure for respectively the p target genes and the q factor genes at time t .

We propose two application frameworks for our modeling. In the first setting, the set of potential factors are the genes expression levels observed at the previous time. Then for all $j \leq q$, for all $t > 1$, $X_t^j = Y_{t-1}^j$ and $q = p$. In that case, we use the Dynamic Bayesian Network (DBN) modeling described in section 3.2. In the second setting, we study the effect of q factor genes differing from the p target genes. Then the effect of factor genes are either simultaneous or time-delayed. We illustrate the latter case in section 4.4 with a real data analysis based on knocked out transcription factors.

Multiple changepoints

Let n be the number of time measurements and $N = \{1, \dots, n\}$ be the set of time measurements. In practice, the time points are not necessary equidistant. Nevertheless, one may consider in that case that the biologist chose time measurements homogeneously in terms of relative reaction rate. We allow the relationships between genes to vary across the process by exploiting a multiple changepoint model. For all target gene i , we introduce an unknown number k^i of changepoints defining $k^i + 1$ non-overlapping phases. Phase $h = 1, \dots, k^i + 1$ starts at changepoint ξ_{h-1}^i and stops before ξ_h^i , where $\xi^i = (\xi_0^i, \dots, \xi_{h-1}^i, \xi_h^i, \dots, \xi_{k^i+1}^i)$ with $\xi_{h-1}^i < \xi_h^i$. To delimitate the bounds, we set $\xi_0^i = 1$ and $\xi_{k^i+1}^i = n + 1$. Thus vector ξ^i has length $|\xi^i| = k^i + 2$. We denote by $\xi = \{\xi^i\}_{i \in P}$ the set of changepoints. Within any phase h , the way target gene i expression depends on the q factor genes expressions is defined by both,

- a set of s_h^i predictors denoted by $\tau_h^i = \{j_1, \dots, j_{s_h^i}\}$, $\tau_h^i \subseteq Q$, $|\tau_h^i| = s_h^i$,
- and a set of parameters $\theta_h^i = ((a_h^{ij})_{j \in \tau_h^i}, \sigma_h^i)$; $a_h^{ij} \in \mathbb{R}$, $\sigma_h^i > 0$. For $j \notin \tau_h^i$, we impose moreover that $a_h^{ij} = 0$.

Let $k = \sum_{i=1}^p k^i$ be the total number of changepoints. We set to \bar{k} the maximal number of changepoints and \bar{s} the maximal number of predictors for each phase, that is we impose $k \leq \bar{k}$ and for all gene i , for all phase h , $s_h^i \leq \bar{s}$.

Regression model

For all target gene i , for all time t , the phase h such that $\xi_{h-1}^i \leq t < \xi_h^i$ determines the way random variable Y_t^i depends on the q factor variables $\{X_t^j; j \in Q\}$ according to the next model,

$$Y_t^i = a_h^{i0} + \sum_{j \in \tau_h^i} a_h^{ij} X_t^j + \varepsilon_t^i, \quad \text{for all } t \text{ such that } \xi_{h-1}^i \leq t < \xi_h^i, \quad (4.1)$$

where $\tau_h^i \subseteq Q$ and $\varepsilon_t^i \sim \mathcal{N}(0, \sigma_h^i)$. We assume moreover that the errors $\{\varepsilon_t^i; 1 \leq i \leq p, 1 \leq t \leq n\}$ are non-correlated.

4.2 Two Steps Reversible Jump MCMC inference

The overall parameter space Θ writes as a finite union of subspaces $\Theta = \bigcup_{k=0}^{\bar{k}} E_k \times \Theta_k$, where

$$E_k = \left\{ \xi = \{\xi^i\}_{i \in P}; \forall i \in P, \xi^i \subseteq \{1, \dots, n+1\}, |\xi^i| = k^i, \xi_0^i = 1, \xi_{k^i+1}^i = n+1, \xi_{h-1}^i < \xi_h^i, \sum_{i=1}^p k^i = k \right\}$$

$$\Theta_k = \prod_{i=1}^p \prod_{h=1}^{k^i} \left\{ \bigcup_{s_h^i=0}^{\bar{s}} \{s_h^i\} \times B_{s_h^i} \right\},$$

with $B_0 = \mathbb{R} \times \mathbb{R}^+$, $B_{s_h^i} = \mathbb{R}^{s_h^i+1} \times \mathcal{P}_{s_h^i}(Q) \times \mathbb{R}^+$ for $k \geq 1$ and $\mathcal{P}_{s_h^i}(Q)$ contains all subsets of Q of dimension s_h^i . We propose to use a reversible jump MCMC procedure to infer posterior distribution of both the topology and the parameters of the time dependent network defined in section 4.1.2. Indeed, Green [Gre95] introduced the reversible jump MCMC method to perform Bayesian computation especially when the dimension of the model remains unknown. The key idea is to construct reversible Markov chain samplers that jump between parameter subspaces of different dimension. This allows to build an ergodic Markov chain whose equilibrium distribution is the desired posterior distribution. The particularity of our approach is to infer simultaneously two ‘‘levels’’ of dimension: the number k of changepoints and the number s_h^i of predictor variables describing gene i expression in each phase h .

To such an aim, we introduce a two steps reversible jump MCMC procedure (summarized in Figure 4.2) which allows to infer posterior density of both changepoints position vector ξ and predictors subsets $\{\tau_h^i\}$ within phases. An estimation of model regression parameters (b, σ) posterior density can be derived as well. Indeed, under weak additional assumptions, the $S \gg 1$ samples generated by the Markov chain $(k^{(r)}, \xi^{(r)}, s^{(r)}, \tau^{(r)}, \theta^{(r)})_{r \in \mathbb{N}}$ are asymptotically distributed according to the posterior distribution $\mathbf{p}(k, \xi, s, \tau, \theta | y)$ and thus allow estimation of all posterior density of interest. For example,

$$\hat{\mathbf{p}}(\xi = \xi' | y) = \frac{1}{S} \sum_{r=1}^S \mathbf{1}_{\{\xi'\}}(\xi^{(r)}) \quad \text{and} \quad \hat{\mathbb{E}}(\theta | y, \xi = \xi') = \frac{\sum_{r=1}^S \theta^{(r)} \mathbf{1}_{\{\xi'\}}(\xi^{(r)})}{\sum_{r=1}^S \mathbf{1}_{\{\xi'\}}(\xi^{(r)})}, \quad (4.2)$$

where y refers to the whole set of observations.

-
1. Initialization: set $(k, \xi, s, \tau, \theta) \in \Theta$.
 2. Iteration i
 - Sample $u \sim \mathcal{U}_{[0,1]}$.
 - If $(u \leq b_k)$
 - then consider changepoint birth (see Fig 4.3),
 - else if $(u \leq b_k + d_k)$ then consider changepoint death (see Fig 4.4),
 - else if $(u \leq b_k + d_k + \pi_k)$ then consider changepoint position change (see Fig 4.5),
 - else consider Regression model change within phases (see Fig 4.6).
 - End If.
 3. $i \leftarrow i + 1$ and go to 2.
-

Figure 4.2: Outline of the algorithm.

Let m^i be the number of repeated measurements of random variable Y_t^i . For all $1 \leq l \leq m^i$, we denote by y_{tl}^i the l^{th} observed value of random variable Y_t^i . Then $y_h^i = (y_{tl}^i)_{\xi_h^i \leq t \leq \xi_{h+1}^i, 1 \leq l \leq m^i}$ is the vector of the $m^i(\xi_h^i - \xi_{h-1}^i)$ observations for gene i in phase h . We denote by $s = \{s_h^i\}_{i \in P, 1 \leq h \leq k^i+1}$ and $\tau = \{\tau_h^i\}_{i \in P, 1 \leq h \leq k^i+1}$ the vector of, respectively, the number and the set of predictors in each phase h of each gene i . There is a natural hierarchical structure to this setup [Gre95], which we formalize by modeling the joint distribution of all variables as the product,

$$\mathbf{p}(k, \xi, s, \tau, \theta, y) = \mathbf{p}_{\bar{k}}(k) \mathbf{p}(\xi|k) \prod_{i=1}^p \prod_{h=1}^{k^i+1} \mathbf{p}(s_h^i, \tau_h^i, \theta_h^i) \mathbf{p}(y_h^i | \xi_{h-1}^i, \xi_h^i, s_h^i, \tau_h^i, \theta_h^i) \quad (4.3)$$

where $\mathbf{p}_{\bar{k}}(k)$ and $\mathbf{p}(\xi|k)$ are the prior probabilities of, respectively, model dimension and parameters for the changepoint vector ξ , $\mathbf{p}(s_h^i, \tau_h^i, \theta_h^i)$ is the prior probabilities of the regression model for phase h of gene i and $\mathbf{p}(y_h^i | \xi_{h-1}^i, \xi_h^i, s_h^i, \tau_h^i, \theta_h^i)$ is the likelihood of the observations of gene i expression level in phase h .

As the number of both changepoints $k = \sum_{i=1}^p k^i$ and factor variables s_h^i within each phase are unknown, we construct a reversible jump MCMC sampler that is directly able to sample from the joint distribution on $\Theta = \cup_{k=0}^{\bar{k}} E_k \times \Theta_k$. Thus we propose candidates according to a set of proposal distributions. These candidates are randomly accepted according to an acceptance ratio that ensures reversibility and thus invariance of the Markov chain with respect to the posterior distribution.

Following Green [Gre95], we propose four different move types (B, D, P, R): birth of a new changepoint, death of an existing changepoint, position change of a changepoint and regression model change within segments. These moves occur with probability b_k for B , d_k for D , π_k for P and η_k for R , depending only on the current number of changepoints k and satisfying $\eta_k + \pi_k + b_k + d_k = 1$. We impose $d_0 = \pi_0 = 0$ and $b_{\bar{k}} = 0$ to preserve the restriction on the number of changepoints. Otherwise, these probabilities are chosen as follows,

$$b_k = c \min \left\{ 1, \frac{\mathbf{p}_{\bar{k}}(k+1)}{\mathbf{p}_{\bar{k}}(k)} \right\}, \quad d_k = c \min \left\{ 1, \frac{\mathbf{p}_{\bar{k}}(k-1)}{\mathbf{p}_{\bar{k}}(k)} \right\} \quad (4.4)$$

where $\mathbf{p}_{\bar{k}}$ is a truncated Poisson density defined in (4.5) and the constant c is chosen smaller than $1/4$ so that the regression models or position changes are more often proposed than changepoint moves. This improve quality of estimation of both changepoint positions and regression models within phases posterior density.

This two-steps reversible jump MCMC procedure allowing the estimation of the posterior density of both changepoint position vector and regression model parameters within phases is summarized in Figure 4.2.

The changepoint birth and death moves represent changes from, respectively, k to $k + 1$ phases and k to $k - 1$ phases. Changepoint position change move is a Metropolis update of an existing changepoint conditional on k changepoints. Finally, regression model change within phases is a second reversible jump MCMC step which is adapted from the approach for model selection by Andrieu and Doucet [AD99]. They introduce an original Bayesian approach where the unknown regression model parameters s_h^i and θ_h^i are regarded as being drawn from appropriate prior distributions.

These four moves allow to generate samples from probability distributions defined on unions of spaces of different dimensions for both the number k of changepoints and the number s_h^i of predictors within each phase. Then we obtain easy evaluation of posterior density for (k, ξ) and conditional posterior density for the regression models structure (s_h^i, τ_h^i) and parameters $(a_h^i, \sigma_h^i)_{1 \leq i \leq p; 1 \leq h \leq k^i}$ given (k, ξ) .

4.2.1 Prior Distributions

Following multiple changepoint approaches involving reversible jump MCMC [Gre95, SWDS03], we assume the number of changepoints k is distributed a priori as a truncated Poisson random variable with mean λ and maximum \bar{k} .

$$\forall k \leq \bar{k}, \quad \mathbf{p}_{\bar{k}}(k) \propto \frac{\lambda^k}{k!} \mathbf{1}_{\{k \leq \bar{k}\}}. \quad (4.5)$$

Conditional on k changepoints, we assume that the changepoint positions vector $\xi = (\xi_0^i, \xi_1^i, \dots, \xi_{k^i+1}^i)_{1 \leq i \leq p}$ takes only integer values which are non-overlapping and uniformly distributed. There are $(n - 1)p$ possible positions for the k changepoints, thus vector ξ has prior density,

$$\mathbf{p}(\xi|k) = 1 / \binom{(n-1)p}{k}. \quad (4.6)$$

For $(s_h^i, \tau_h^i, \theta_h^i)$, we use the priors proposed by Andrieu et Doucet [AD99] in their reversible jump MCMC approach for model selection. So we assume the structure,

$$\mathbf{p}(s_h^i, \tau_h^i, \theta_h^i) = \mathbf{p}(s_h^i, \tau_h^i, a_h^i | \sigma_h^i) \mathbf{p}(\sigma_h^i). \quad (4.7)$$

where $a_h^i = (a_h^{ij})_{j \in Q}$ and σ_h^i is a scale parameter that is assumed to be distributed according to a conjugate inverse-Gamma prior distribution ($(\sigma_h^i)^2 \sim \mathcal{IG}(v_0/2, \gamma_0/2)$). As recommended in [AD99], we chose $v_0 = 0$ and $\gamma_0 = 0$ to obtain Jeffrey's uninformative prior $\mathbf{p}((\sigma_h^i)^2) \propto 1/(\sigma_h^i)^2$ and the prior distribution,

$$\mathbf{p}(s_h^i, \tau_h^i, a_h^i | \sigma_h^i) \propto \mathbf{p}_{\bar{s}}(s_h^i) \mathbf{p}(\tau_h^i | s_h^i) \mathbf{p}(a_h^i | \sigma_h^i, s_h^i, \tau_h^i). \quad (4.8)$$

The prior probability model distribution is a truncated Poisson distribution $\mathbf{p}_{\bar{s}}(s_h^i)$ with mean Λ and maximum \bar{s} . The sets of predictors τ_h^i are assumed uniformly distributed conditional on $|\tau_h^i| = s_h^i$,

$$\forall \tau \subseteq Q, \quad |\tau| = s, \quad \mathbf{p}(\tau | s) = 1 / \binom{Q}{s}. \quad (4.9)$$

Conditional on factor genes set τ_h^i of size s_h^i , the $s_h^i + 1$ regression coefficients, denoted by $a_{\tau_h^i}^i = (a_h^{i0}, (a_h^{ij})_{j \in \tau_h^i})$, are assumed zero-mean Gaussian with covariance $(\sigma_h^i)^2 \Sigma_{\tau_h^i}^i$,

$$\mathbf{p}(a_h^i | \sigma_h^i, s_h^i, \tau_h^i) = |2\pi(\sigma_h^i)^2 \Sigma_{\tau_h^i}^i|^{-1/2} \exp \left[-\frac{a_{\tau_h^i}^i \Sigma_{\tau_h^i}^{-1} a_{\tau_h^i}^i}{2(\sigma_h^i)^2} \right], \quad (4.10)$$

where $\Sigma_{\tau_h^i} = \delta^{-2} D_{\tau_h^i}^t(x) D_{\tau_h^i}(x)$ and $D_{\tau_h^i}(x)$ is the $m^i(\xi_h^i - \xi_{h-1}^i) \times (s_h^i + 1)$ matrix whose first column is a vector of 1 (for the constant in model (4.1)) and each $j + 1^{\text{th}}$ column contains the observed (eventually repeated) value $(x_{tl}^j)_{\xi_{h-1}^i \leq t < \xi_h^i; 1 \leq l \leq m}$ for all factor gene j in τ_h^i .

The terms λ and Λ can be interpreted as the expected number of changepoints and predictor variables respectively and the term δ^2 as the expected signal-to-noise ratio. Following [AD99], these parameters are drawn according to uninformative conjugate priors: $\lambda, \Lambda \sim \mathcal{Ga}(1/2 + \varepsilon_1, \varepsilon_2)$ with $\varepsilon_i \ll 1$ for $i = 1, 2$ and $\delta^2 \sim \mathcal{IG}(\alpha_{\delta^2}, \beta_{\delta^2})$ with $\alpha_{\delta^2} = 2$ and $\beta_{\delta^2} > 0$.

Finally, conditional on ξ^i , the likelihood of the observations y_h^i for target gene i in phase h writes,

$$\mathbf{p}(y_h^i | \xi_{h-1}^i, \xi_h^i, s_h^i, \tau_h^i, \theta_h^i) = (2\pi(\sigma_h^i)^2)^{-\frac{m^i(\xi_h^i - \xi_{h-1}^i)}{2}} \exp \left\{ -\frac{1}{2(\sigma_h^i)^2} (y_h^i - D_{\tau_h^i}(x) a_{\tau_h^i}^i)^t (y_h^i - D_{\tau_h^i}(x) a_{\tau_h^i}^i) \right\}. \quad (4.11)$$

4.2.2 Parameter space posterior distribution: integration of the ‘‘nuisance’’ parameters (a, σ)

From Bayes theorem, parameters $(k, \xi, s, \tau, a, \sigma | y)$ posterior density satisfies,

$$\mathbf{p}(k, \xi, s, \tau, a, \sigma | y) \propto \mathbf{p}(k) \mathbf{p}(\xi | k) \prod_{i=1}^p \prod_{h=1}^{k^i} \mathbf{p}(y_h^i | \xi_{h-1}^i, \xi_h^i, s_h^i, \tau_h^i, a_h^i, \sigma_h^i) \mathbf{p}(s_h^i, \tau_h^i, a_h^i | \sigma_h^i) \mathbf{p}(\sigma_h^i). \quad (4.12)$$

Following the approach proposed by Andrieu and Doucet [AD99] for model selection, we carry out integration of the ‘‘nuisance’’ parameters (a, σ) in order to obtain an expression of posterior density $\mathbf{p}(k, \xi, s, \tau | y)$ up to a normalization constant. Thus we can consider a changepoint move proposal without previously generating regression model parameters (a_h^*, σ_h^*) for the modified phases. Moreover, the acceptance probability of the move does not depend on simulated parameters (a_h^*, σ_h^*) . From equations (4.11, 4.10, 4.6, 4.5), left hand side of equation (4.12) is entirely explicit and we have,

$$\begin{aligned} \mathbf{p}(y_h^i | \xi_{h-1}^i, \xi_h^i, s_h^i, \tau_h^i, a_h^i, \sigma_h^i) \mathbf{p}(s_h^i, \tau_h^i, a_h^i | \sigma_h^i) \mathbf{p}(\sigma_h^i) = \\ \frac{\Lambda^{k_h^i}}{norm_{\Lambda}} \frac{(q - s_h^i)!}{q!} \frac{1}{|2\pi(\sigma_h^i)^2 \Sigma_{\tau_h^i}|^{1/2}} \exp \left\{ -\frac{1}{2(\sigma_h^i)^2} (a_h^i - d_h^i)^t (M_h^i)^{-1} (a_h^i - d_h^i) \right\} \\ \exp \left\{ -\frac{1}{2(\sigma_h^i)^2} (v_0 + (y_h^i)^t P_h^i y_h^i) \right\} (\sigma_h^i)^{-(v_0 + m^i(\xi_h^i - \xi_{h-1}^i))/2} (2\pi)^{-m^i(\xi_h^i - \xi_{h-1}^i)/2} \end{aligned} \quad (4.13)$$

where $norm_{\Lambda}$ is the normalization for the truncated Poisson distribution and matrices P_h^i , M_h^i and vector d_h^i are defined as follows, with I referring to the identity matrix of size $m^i(\xi_h^i - \xi_{h-1}^i)$,

$$P_h^i = I - D_{\tau_h^i}(x) M_h^i D_{\tau_h^i}^t(x), \quad (4.14)$$

$$M_h^i = \frac{\delta^2}{\delta^2 + 1} \left(D_{\tau_h^i}^t(x) D_{\tau_h^i}(x) \right)^{-1}, \quad (4.15)$$

$$d_h^i = M_h^i D_{\tau_h^i}^t(x) y_h^i. \quad (4.16)$$

-
- Propose a new changepoint position at random in $\{2, \dots, n\}^p \setminus \{\xi\}$.
 - Propose the side (right or left) at random for the new phase.
 - Update λ from (4.31) given the number $s_{h^*}^i$ of predictors in the old phase h^* .
 - Draw a new set of predictors (s^*, τ^*) from equations (4.5, 4.9) for the new phase.
 - Compute $A_{k,k+1}(\xi, \xi^+)$ from (4.22) and sample $u \sim \mathcal{U}_{[0,1]}$.
 If $u \leq A_{k,k+1}(\xi, \xi^+)$, then sample (a^*, σ^*) from (4.28, 4.29) and the state of the Markov chain becomes $(k+1, \xi^+, s^+, \tau^+, a^+, \sigma^+)$,
 else it remains $(k, \xi, s, \tau, a, \sigma)$.
-

Figure 4.3: Changepoint birth move (B).

The integration of a_h^i (normal distribution) and then of σ_h^i (inverse gamma distribution) yields,

$$\mathbf{p}(k, \xi, s, \tau | y) \propto \frac{((n-1)p-k)!}{((n-1)p)!} \lambda^k \left(\frac{(\frac{\gamma_0}{2})^{v_0/2}}{\Gamma(\frac{v_0}{2}) \text{norm}_\Lambda} \right)^{p+k} (2\pi)^{-\frac{n}{2} \sum_{i=1}^p m^i} \prod_{i=1}^p \prod_{h=1}^{k^i} \left\{ \frac{(q-s_h^i)! \Lambda^{s_h^i}}{q! (\delta^2+1)^{(s_h^i+1)/2}} \Gamma\left(\frac{v_0+m^i(\xi_h^i-\xi_{h-1}^i)}{2}\right) \left(\frac{v_0+(y_h^i)^t P_h^i y_h^i}{2}\right)^{-\frac{v_0+m^i(\xi_h^i-\xi_{h-1}^i)}{2}} \right\}. \quad (4.17)$$

We use this posterior distribution for proposed transitions in the changepoint moves: birth (B), death (D) or position change (P).

4.2.3 Four moves

We form the reversible jump MCMC acceptance probability for changepoint birth move as $\min\{1, r_{k,k+1}\}$ where,

$$r_{k,k+1} = (\text{posterior distribution ratio}) \times (\text{proposal ratio}) \times (\text{Jacobian}). \quad (4.18)$$

Changepoint death move is respectively accepted with probability $\min\{1, r_{k,k-1}\}$. To calculate the proposal ratio and Jacobian in (4.18), we begin by describing our changepoint proposals. The different moves are defined by heuristic considerations, the only consideration to be fulfilled being to maintain the correct invariant distribution of the Markov chain. As pointed out in [ADD01], a particular choice of move proposal will only influence on the convergence rate of the algorithm; nevertheless the proposed ones led to satisfactory results.

Birth of a changepoint

Let $\xi = \{\xi_0^i, \dots, \xi_{k^i+1}^i; 1 \leq i \leq p\}$ be the current changepoints vector in the model. Changepoint birth move is summarized in Figure 4.3. When considering birth of a new changepoint, we first draw a new integer changepoint position ξ^* uniformly along the sites in the sequences that do not currently contain changepoints,

$$\xi^* | \xi \sim \mathcal{U}_{\{2, \dots, n\}^p \setminus \{\xi\}}. \quad (4.19)$$

The new changepoint is within a current phase h^* of a target gene i such that $\xi_{h^*-1}^i < \xi^* < \xi_{h^*}^i$. This phase starts at changepoint $\xi_{h^*-1}^i$ and ends at $\xi_{h^*}^i - 1$, where $\xi_0^i = 1$ and $\xi_{k^i+1}^i = n+1$ as previously defined. The proposed new changepoint vector is $\xi^+ = \xi \cup \{\xi^*\}$.

-
- Propose an existing changepoint ξ_h^{i*} at random in $\{\xi_h^i; 1 \leq i \leq p, 1 \leq h \leq k^i\}$.
 - Sample the set of predictors for the collapsed phase τ^* from $(\tau_h^{i*}, \tau_{h+1}^{i*})$.
 - Compute $A_{k,k-1}(\xi, \xi^-)$ from (4.23) and sample $u \sim \mathcal{U}_{[0,1]}$.
 If $u \leq A_{k,k-1}(\xi, \xi^-)$,
 then the state of the Markov chain becomes $(k-1, \xi^-, s^-, \tau^-, a^-, \sigma^-)$,
 else it remains $(k, \xi, s, \tau, a, \sigma)$.
-

Figure 4.4: Changepoint death move (D).

So the proposal ratio writes,

$$\frac{d_{k+1} q(\xi|\xi^+)}{b_k q(\xi^+|\xi)} = \frac{(n-1)p-k}{\lambda}, \quad (4.20)$$

where $q(\xi^+|\xi) = 1/((n-1)p-k)$ is the probability of drawing new changepoint ξ^+ when adding an extra changepoint to vector ξ (respectively $q(\xi|\xi^+) = 1/(k+1)$ when deleting one changepoint of vector ξ^+).

We divide phase h^* into two new segments h_L^* and h_R^* for the left and right side of the old phase h^* respectively. Following Suchard et al. [SWDS03], the set of predictors τ_{h^*} of the selected “old” phase h^* is attributed to the left hand new phase h_L^* with probability 1/2; to the right hand new phase h_R^* otherwise. For the remaining new phase, we draw a new set of predictors (s^*, τ^*) from equations (4.5, 4.9) after updating λ from (4.31) given the number of predictors $s_{h^*}^i$ in the previous phase h^* .

The Jacobian equals 1. The proposal ratio is made explicit in equation (4.20). Let $s^+ = s \cup \{s^*\}$ and $\tau^+ = \tau \cup \{\tau^*\}$ define the sets of predictors of the phases delimited by the proposed new changepoints vector ξ^+ . The posterior distribution ratio $\frac{\mathbf{p}(k+1, \xi^+, s^+, \tau^+ | y)}{\mathbf{p}(k, \xi, s, \tau | y)}$ is computed from equation (4.17). Thus we obtain,

$$r_{k,k+1}(\xi, \xi^+) = \frac{\left(\frac{\gamma_0}{2}\right)^{v_0/2}}{\Gamma\left(\frac{v_0}{2}\right) \text{norm}_\Lambda} \frac{(q-s^*)! \Lambda^{s^*}}{q!(\delta^2+1)^{(s^*+1)/2}} \frac{\Gamma_{h_L^*} \Gamma_{h_R^*}}{\Gamma_{h^*}} \left(\frac{v_0 + (y_{h^*}^i)^t P_{h^*}^i y_{h^*}^i}{2}\right)^{\frac{1}{2}(v_0+m^i(\xi_{h^*}^i - \xi_{h^*-1}^i))} \\ \left(\frac{v_0 + (y_{h_L^*}^i)^t P_{h_L^*}^i y_{h_L^*}^i}{2}\right)^{-\frac{1}{2}(v_0+m^i(\xi_{h_L^*}^i - \xi_{h_L^*-1}^i))} \left(\frac{v_0 + (y_{h_R^*}^i)^t P_{h_R^*}^i y_{h_R^*}^i}{2}\right)^{-\frac{1}{2}(v_0+m^i(\xi_{h_R^*}^i - \xi_{h_R^*-1}^i))}. \quad (4.21)$$

where, for all h in $\{1, \dots, k^i+1\}$, $\Gamma_h = \Gamma\left(\frac{v_0+m^i(\xi_h^i - \xi_{h-1}^i)}{2}\right)$. The birth of the proposed changepoint is accepted with probability,

$$\alpha_{k,k+1}(\xi, \xi^+) = \min\{1, r_{k,k+1}(\xi, \xi^+)\}. \quad (4.22)$$

If birth is accepted, we sample new parameters (a^*, σ^*) from (4.28, 4.29) for the new phase described by (s^*, τ^*) . Then the state of the Markov chain becomes $(k+1, \xi^+, s^+, \tau^+, a^+, \sigma^+)$, where $a^+ = a \cup \{a^*\}$ and $\sigma^+ = \sigma \cup \{\sigma^*\}$. Otherwise the Markov chain remains unchanged.

Death of a changepoint

When considering death of an existing changepoint (summarized in Figure 4.4), we first draw an existing changepoint ξ_h^{i*} uniformly from $\{\xi_h^i; 1 \leq i \leq p, 1 \leq h \leq k^i\}$ to collapse the neighboring phases of changepoint ξ_h^{i*} and thus form a single phase. Then the proposed new changepoint

-
- Propose an existing changepoint ξ_h^{i*} at random in $\{\xi_h^i; 1 \leq i \leq p, 1 \leq h \leq k^i\}$.
 - Draw a new position uniformly from $[\xi_h^{i*} - W/2, \xi_h^{i*} + W/2] \cap [\xi_{h-1}^{i*} + 1, \xi_{h+1}^{i*} - 1]$.
 - Compute $\alpha_k(\xi, \xi^*)$ from (4.25) and sample $u \sim \mathcal{U}_{[0,1]}$.
If $u \leq \alpha_k(\xi, \xi^*)$, then the state of the Markov chain becomes $(k, \xi^*, s, \tau, a, \sigma)$,
else it remains $(k, \xi, s, \tau, a, \sigma)$.
-

Figure 4.5: Position change move (P).

vector is $\xi^- = \xi \setminus \{\xi_h^{i*}\}$. The set τ^* of predictors for the new phase is τ_h^{i*} with probability $1/2$; τ_{h+1}^{i*} otherwise. The death acceptance probability is,

$$\alpha_{k,k-1}(\xi, \xi^-) = \min\{1, r_{k-1,k}^{-1}(\xi^-, \xi)\}. \quad (4.23)$$

If death is accepted the neighboring phases are collapsed and the state of the Markov chain becomes $(k-1, \xi^-, s^-, \tau^-, a^-, \sigma^-)$ where the parameters having the superscript $-$ have been reduced from the deleted phase parameters, otherwise the Markov chain remains unchanged.

Position change of a changepoint

Changepoint position change move is a Metropolis update summarized in Figure 4.5. We first choose an existing changepoint ξ_h^i uniformly from $\{\xi_h^i; 1 \leq i \leq p, 1 \leq h \leq k^i\}$. Then we propose to update changepoint ξ_h^i by drawing a new position ξ_h^{i*} uniformly from $[\max\{\xi_h^i - W/2, \xi_{h-1}^i + 1\}, \xi_h^i - 1] \cup [\xi_h^i + 1, \min\{\xi_h^i + W/2, \xi_{h+1}^i - 1\}]$, where W is a tunable window size. For short time series, we choose $W = 2$. The new changepoint vector ξ^* , obtained by replacing ξ_h^i with ξ_h^{i*} , is accepted with probability $\alpha_k(\xi, \xi^*) = \min\{1, r_k(\xi, \xi^*)\}$ where,

$$r_k(\xi, \xi^*) = \frac{\mathbf{p}(k, \xi^*, s, \tau | y) q(\xi | \xi^*)}{\mathbf{p}(k, \xi, s, \tau | y) q(\xi^* | \xi)}, \quad (4.24)$$

and $q(\xi^* | \xi)$ is the probability of drawing new changepoint ξ^* when changing position of one changepoint of vector ξ . The number of new changepoints vectors that can be proposed by changing position of one element of vector ξ is equal to $kW - e$ where e is the number of impossible position changes because the gap between two successive changepoints is smaller than $W/2$. Then,

$$r_k(\xi, \xi^*) = \left(\frac{(v_0 + (y_h^{i*})^t P_h^{i*} y_h^{i*})^{v_0+m^i(\xi_h^{i*}-\xi_{h-1}^i)} (v_0 + (y_{h+1}^{i*})^t P_{h+1}^{i*} y_{h+1}^{i*})^{v_0+m^i(\xi_{h+1}^i-\xi_h^{i*})}}{(v_0 + (y_h^i)^t P_h^i y_h^i)^{v_0+m^i(\xi_h^i-\xi_{h-1}^i)} (v_0 + (y_{h+1}^i)^t P_{h+1}^i y_{h+1}^i)^{v_0+m^i(\xi_{h+1}^i-\xi_h^i)}} \right)^{1/2} \frac{\Gamma\left(\frac{v_0+m^i(\xi_h^{i*}-\xi_{h-1}^i)}{2}\right) \Gamma\left(\frac{v_0+m^i(\xi_{h+1}^i-\xi_h^{i*})}{2}\right)}{\Gamma\left(\frac{v_0+m^i(\xi_h^i-\xi_{h-1}^i)}{2}\right) \Gamma\left(\frac{v_0+m^i(\xi_{h+1}^i-\xi_h^i)}{2}\right)} \frac{kW - e}{kW - e^*}, \quad (4.25)$$

where y_h^{i*} and P_h^{i*} respectively refer to gene i observations and projection matrix in phase h of changepoint ξ^* .

Regression model change within phases

For regression model change move, we use a second level of reversible jump MCMC computations based on the model selection procedure by Andrieu and Doucet [AD99, ADD01]. When this move is chosen, we consider regression model change within all current phases. So for all phase h of

For all target gene i , for all phase h , $1 \leq h \leq k^i$,

- update hyperparameters δ, λ according to (4.32,4.31),
- sample $u \sim \mathcal{U}_{[0,1]}$.
- If $u \leq b_{s_h^i}$, then consider birth of a predictor in phase h ,
 else if $u \leq b_{s_h^i} + d_{s_h^i}$ then consider death of a predictor
 else update parameters $(a_{\tau_h^i}^i, \sigma_h^i)$ according to (4.28,4.29).

Figure 4.6: Regression model change (R).

all target gene i successively, we propose three different moves: birth of a new predictor, death of an existing predictor or update of regression model parameters $(a_{\tau_h^i}, \sigma_h^i)$. The predictor birth and death moves represent changes from, respectively, s_h^i to $s_h^i + 1$ or $s_h^i - 1$ predictors in the regression model. The probability for choosing these moves, respectively $b_{s_h^i}$, $d_{s_h^i}$ and $\eta_{s_h^i}$, satisfy $b_{s_h^i} + d_{s_h^i} + \eta_{s_h^i} = 1$ and are defined as follows,

$$b_{s_h^i} = c_R \min \left\{ 1, \frac{\mathbf{p}_{\bar{s}}(s_h^i + 1)}{\mathbf{p}_{\bar{s}}(s_h^i)} \right\} \quad \text{and} \quad d_{s_h^i} = c_R \min \left\{ 1, \frac{\mathbf{p}_{\bar{s}}(s_h^i - 1)}{\mathbf{p}_{\bar{s}}(s_h^i)} \right\}. \quad (4.26)$$

We take $c_R = 0,5$ so that predictor birth or death moves are often proposed. This allows to range over the set of all possible model structures. When considering predictor birth, a new predictor is uniformly drawn from $\{1, \dots, q\} \setminus \{\tau_h^i\}$ and we set the new predictors subset $\tau_h^{i+} = \tau_h^i \cup \{j^*\}$. Predictor birth move, that is change from τ_h^i to τ_h^{i+} , is accepted with acceptance probability $\alpha_{s_h^i, s_h^i+1}(\tau_h^i, \tau_h^{i+}) = \min\{1, r_{s_h^i, s_h^i+1}(\tau_h^i, \tau_h^{i+})\}$ where,

$$r_{s_h^i, s_h^i+1}(\tau_h^i, \tau_h^{i+}) = \frac{1}{\sqrt{1 + \delta^2}} \left(\frac{\gamma_0 + (y_h^i)^t P_{\tau_h^i} y_h^i}{\gamma_0 + (y_h^i)^t P_{\tau_h^{i+}} y_h^i} \right)^{(m^i(\xi_h^i - \xi_{h-1}^i) + \nu_0)/2}. \quad (4.27)$$

Computation of $r_{s_h^i, s_h^i+1}(\tau_h^i, \tau_h^{i+})$ is carried out by following Andrieu and Doucet [AD99]; see Appendix 4.5.2 for details. In the same manner, predictor death move is accepted with probability $\alpha_{s_h^i, s_h^i-1}(\tau_h^i, \tau_h^{i-}) = \min\{1, r_{s_h^i-1, s_h^i}^{-1}(\tau_h^i, \tau_h^{i-})\}$. The update of regression model parameters is computed from equations (4.28, 4.29). The outline of regression model change move (R) is summarized in Figure 4.6 and the complete development of each move in Figure 4.7.

Updating hyperparameters

Parameter λ is updated at each iteration of this 2-step reversible jump MCMC procedure and parameters (δ^2, Λ) each time regression model change moves (R) within phases is chosen. Updating is carried out as follows,

$$\lambda \mid k \sim \mathcal{G}a\left(\frac{1}{2} + k + \varepsilon_1, 1 + \varepsilon_2\right), \quad (4.30)$$

$$\Lambda \mid s_h^i \sim \mathcal{G}a\left(\frac{1}{2} + s_h^i + \varepsilon_1, 1 + \varepsilon_2\right), \quad (4.31)$$

$$\delta \mid s_h^i, \tau_h^i, \theta_{\tau_h^i} \sim \mathcal{IG} \left(s_h^i + \alpha_{\delta^2}, \frac{a_{\tau_h^i} D_{\tau_h^i}^t(x) D_{\tau_h^i}(x) a_{\tau_h^i}}{2(\sigma_h^i)^2} + \beta_{\delta^2} \right). \quad (4.32)$$

Predictor birth

- Choose a new predictor $j^* \sim \mathcal{U}_{\{1, \dots, q\} \setminus \{\tau_h^i\}}$ and set $\tau_h^{i+} = \tau_h^i \cup \{j^*\}$.
- Compute $\alpha_{s_h^i, s_h^i+1}(\tau_h^i, \tau_h^{i+}) = \min\{1, r_{s_h^i, s_h^i+1}(\tau_h^i, \tau_h^{i+})\}$ from (4.27).
- Sample $u \sim \mathcal{U}_{\{0,1\}}$.
If $u \leq \alpha_{s_h^i, s_h^i+1}(\tau_h^i, \tau_h^{i+})$ then the model in phase h becomes $(s_h^i + 1, \tau_h^{i+})$,
else the model remains unchanged: update parameters $(a_{\tau_h^i}^i, \sigma_h^i)$ according to equations (4.28, 4.29).

Predictor death

- Choose one predictor $j^* \sim \mathcal{U}_{\{\tau_h^i\}}$ and set $\tau_h^{i-} = \tau_h^i \setminus \{j^*\}$
- Compute $\alpha_{s_h^i, s_h^i-1}(\tau_h^i, \tau_h^{i-}) = \min\{1, r_{s_h^i, s_h^i-1}(\tau_h^i, \tau_h^{i-})\}$ from (4.27).
- Sample $u \sim \mathcal{U}_{\{0,1\}}$.
If $u \leq \alpha_{s_h^i, s_h^i-1}(\tau_h^i, \tau_h^{i-})$ then the state i becomes $(s_h^i - 1, \tau_h^{i-})$,
else the model remains unchanged: update parameters $(a_{\tau_h^i}^i, \sigma_h^i)$ according to equations (4.28, 4.29).

Update predictor parameters

$$(\sigma_h^i)^2 | y_h^i, \tau_h^i \sim \mathcal{IG} \left(\frac{v_0 + m^i(\xi_{h+1}^i - \xi_h^i)}{2}, \frac{\gamma_0}{2} + \frac{1}{2}(y_h^i)^t P_{\tau_h^i} y_h^i \right) \quad (4.28)$$

$$a_{\tau_h^i}^i | y_h^i, \tau_h^i, \sigma_h^i \sim \mathcal{N} \left(\frac{\delta^2}{\delta^2 + 1} \left(D_{\tau_h^i}^t(x) D_{\tau_h^i}(x) \right)^{-1} D_{\tau_h^i}^t(x) y_h^i, \frac{\delta^2 (\sigma_h^i)^2}{\delta^2 + 1} \left(D_{\tau_h^i}^t(x) D_{\tau_h^i}(x) \right)^{-1} \right) \quad (4.29)$$

Figure 4.7: Detailed moves for Regression model change.

4.3 Simulation study

We investigate the accuracy of this two-steps MCMC inference procedure to recover the change-points and edges of multiple changepoint regression models defined by non homogeneous networks. We randomly generated 1000 time series of length $n=50$ and then 1000 according to the multiple changepoint regression models defined in subsection 4.1.2. We set the maximal number of change-points to $s_{max} = 2$ and the maximal number of predictors to $k_{max} = 5$ in the simulation model. Change-points position and predictors set are drawn uniformly given the number of change-points s and given the number of predictors k respectively. Regression coefficients are drawn uniformly as follows: $\sigma^i \sim \mathcal{U}(0.03; 0.08)$, $b^{i0} \sim \mathcal{U}([-2, -0.5] \cup [0.5; 2])$, and, for all $j \neq 0$, $b^{ij} \sim \mathcal{U}([0.2 + \sigma^i; 1 + \sigma^i])$. Values close to 0 are excluded from b^{i0} simulation interval so that the effect of the predictors are detectable. We computed the Positive Predictive Value (PPV) and sensitivity for both change-points and edges detection, the False Positive (FP) rate for change-points detection and the specificity for edges detection.

$$PPV = \frac{TP}{TP + FP} \quad Sensitivity = \frac{TP}{TP + FN} \quad Specificity = \frac{TN}{FP + TN}$$

Figure 4.8 displays the average results for change-points detection according to the number of change-points in the simulation model. The results for edges detection according to the number of edges in the simulation model appear in Figure 4.9.

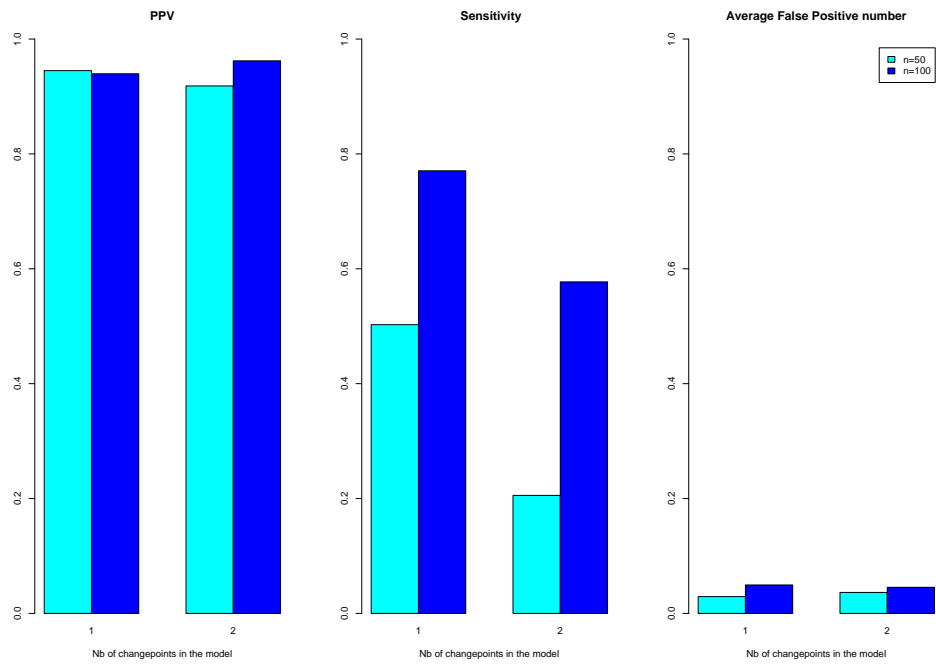


Figure 4.8: Changepoints detection results: Positive Predictive Value, Sensitivity and False Positive rate according to the number of changepoints in the simulation model.

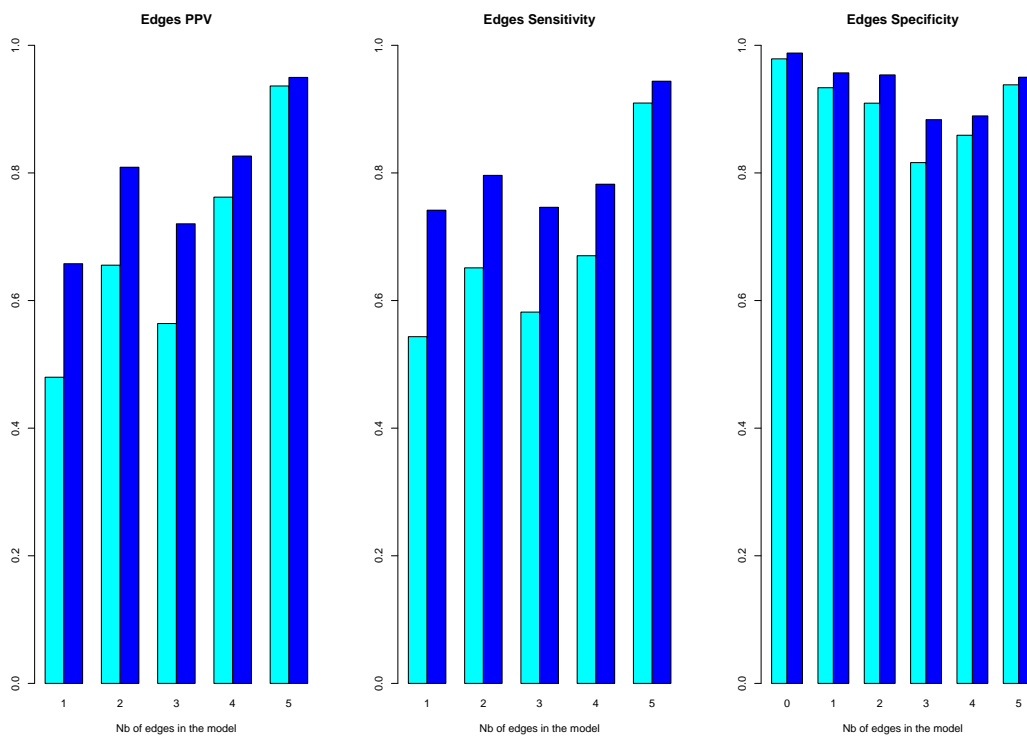


Figure 4.9: Edges detection within phases results: Positive Predictive Value, Sensitivity and Specificity according to the number of edges in the simulation model.

For changepoints detection, the PPV is very high in every situation. The detected changepoints are changepoints in the simulation in more than 90 % of the cases. However the sensitivity is high only if the time series are long enough. When there is one single changepoint in the simulation model, the sensitivity reaches almost 80 % when the length of the series is $n = 100$. However, when the average length of the phase in the simulation model is close to 30 (1 changepoint for $n = 50$ or 2 changepoints for $n = 100$), the sensitivity only reaches 55%. When no repeated time measurements are available, the number of time measurements within each phase has to be high enough so that a changepoint is detected.

As for edges detection, the PPV and sensitivity results are similar and do not clearly depend on the number of edges in the simulation model. The specificity is very high in every cases though.

4.4 Analysis of a Microarray Time Course Data Set

We propose to use this two-steps MCMC procedure to analyze the reaction to the antimetabolic drug benomyl addition by the yeast *S. cerevisiae*. Indeed, in a previous study by Lucau-Danila et al. [LDLK⁺05], the response developed by the yeast was shown to be time delayed and to potentially require the successive activity of several TFs. This is a very interesting application framework to evaluate the pertinence of our model and inference method.

4.4.1 Early genomic response following benomyl addition data

The variety of environmental stresses is probably the major challenge imposed on the transcriptional machinery. To precisely describe the very early genomic response developed by the yeast *S. cerevisiae* to accommodate a chemical stress, Lucau-Danila et al. [LDLK⁺05] performed time course analysis of its genes expression alteration immediately after the addition of benomyl. Parallel experiments were conducted in different genetic contexts. Six genes coding for transcription factors (TFs) supposed to be connected to the drug response were successively deleted: Yap1, Pdr1, Pdr3, Yrr1, Pdr8 and Yrm1. From this experiment, Lucau-Danila et al. pointed out that two well-known key mediators of stress tolerance, Yap1 and Pdr1, appeared to be responsible for the very rapid establishment of a transient transcriptional response encompassing a subset of 119 genes. Even though Yap1 plays a predominant role in this response, this TF does bind, in vivo, promoters of genes which are not automatically up-regulated. So they proposed that Yap1 nuclear localization and DNA binding are necessary but not sufficient to elicit the specificity of the chemical stress response.

Our objective is to go deeper into the analysis of this data in order to determine which other TFs (in addition to YAP1) are implicated in the reaction to benomyl and how their activities are coordinated. A substantial alteration in target genes expression profiles was observed by only four of out of the six different knockout strains: YAP1, PDR1, PDR3 and YRR1 deleted strains. So we study the effect of knocking out the genes coding for these four TFs that we number from 1 to 4. We observe the target gene expression under 5 different conditions: the wild type strain and the strains having one deleted TF out of the four under study. The expression level of each target gene was measured at 5 successive time points after benomyl addition $N = \{+30 \text{ s}, +2 \text{ min}, +4 \text{ min}, +10 \text{ min}, +20 \text{ min}\}$ for the 5 strains. Thus, for each target gene i and for all t in N , we observe an expression measure of the variable Y_t^i in five different conditions ($l = 1 \dots 5$, see Table 4.4.1).

We propose to analyze the effect of each TF knockout on the target genes expression according to the time by basing on the multiple changepoint model described in subsection 4.1.2. Then for

all TF j in $Q = \{1, \dots, 4\}$, for all condition l in $\{1, \dots, 5\}$, the variable X_t^j is set to $x_{tl}^j = 1$ for all t in N if TF j is knocked in condition l , and to $x_{tl}^j = 0$ otherwise. Table 4.4.1 summarizes the data observed for target gene i and the four TFs at each time point. In this study, the values of the predictor variables are set by the experimental design.

l	Y_t^i	X_t^1	X_t^2	X_t^3	X_t^4	Knockout
1	y_{t1}^i	0	0	0	0	\emptyset
2	y_{t2}^i	1	0	0	0	YAP1
3	y_{t3}^i	0	1	0	0	PDR1
4	y_{t4}^i	0	0	1	0	PDR3
5	y_{t5}^i	0	0	0	1	YRR1

Table 4.1: Data collected for target gene i at time t , for all t in N .

4.4.2 First analysis

Approach

We do not have repeated measurements for each time point. Indeed, the expression level of each gene is measured only once in each condition. So we propose to select, for each target genes, the two other target genes having the most similar expression profile over the 5 conditions (in term of Euclidean distance). We build this way groups of 3 genes (see FLR1 and its 2 nearest neighbors expression profiles in Figure 4.10). Thus we assume the genes of a group are regulated by the same TFs and consider their expression profiles as repetition of the same process. This allows to carry out estimation when no time point repeated measurements are available. But we have to point out that this analysis is global: we only derive a global conclusion for the target gene and its 2 nearest neighbors. The approach is acceptable for genes having very similar neighbors (like FLR1 in Figure 4.10). But we can not conclude for the target genes whose expression is similar to no other target gene (like SNG1 in Appendix, Figure 4.31) unless we had repeated measurements.

In order to evaluate the pertinence of this approach, we propose to carry out a first study on four target genes. These genes code for proteins known for being implicated in drug reaction: FLR1, GTT2, TPO1 and SNG1. Proteins FLR1 and TPO1 are multidrug transporters. SNG1 is a protein involved in nitrosoguanidine resistance and GTT2 is a transferase known for being regulated by YAP1. The data collected for the the target genes appear in Appendix (subsections 4.5.3 to 4.5.6). In each subsection, the first figure displays the expression profiles of the target gene and its two nearest neighbors over the five strains under study. The name of the three genes appear in the bottom right hand panel, the first name is the one of the gene coding for the studied protein; then appear the name of the two nearest neighbors, ordered by decreasing similarity.

As we focus on TFs deletion effect, we want to detect a changepoint when the effect of the deletion of a TF starts or ends, that is when a knockout profile starts or ends to differ from the wild type expression profile. Thus we don't want to detect a changepoint when the wild type expression level differs from the one observed at the previous time. Nevertheless, most of the target genes under study have a wild type expression profile that is varying across time. Indeed, the reaction to the stress brought about by benomyl addition is not observed instantaneously but after a short period. Consequently, most of the wild type expression profiles are increasing

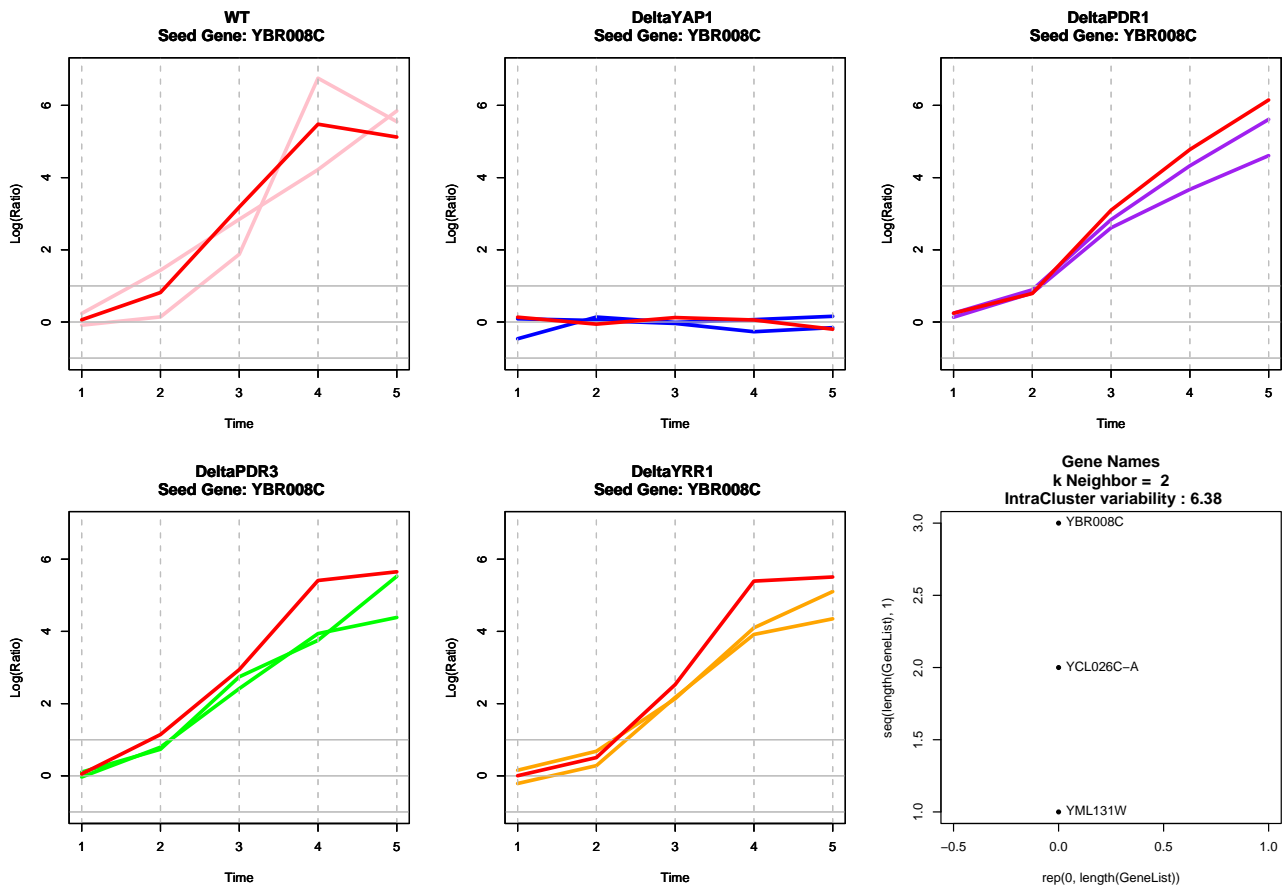


Figure 4.10: Expression profiles of gene coding for protein FLR1 (red plotted) and its 2 nearest neighbors over the 5 strains under study (from left to right: wild type, and successively YAP1, PDR1, PDR3 and YRR1 deleted). FLR1 is membrane multidrug transporter known for being implicated in the answer to benomyl. This protein is early regulated by YAP1.

curves. So, for each group of three genes, for each time point t , we subtract the mean of the wild type (condition 1) expression levels $\frac{1}{3} \sum_{i=1}^3 y_{t1}^i$ observed at time t from each expression level y_{tl}^i measured for all target gene i in all condition l . Then, for all target gene i , for all phase h , we model the corrected expression levels as follows,

$$\forall t \in N, \quad Y_t^i = \sum_{j=1}^q a_h^{ij} X_t^j + \varepsilon_t^i, \quad \varepsilon_t^i \sim \mathcal{N}(0, \sigma^i). \quad (4.33)$$

The effect of TF j deletion in phase h is measured by coefficient a_h^{ij} . By using our 2-steps reversible jump MCMC procedure, we propose to infer a time-dependent network which allows to both point out which TF knockout alter the target genes expression and determine precisely when this alteration occurs. This allows to recover a network describing the chronology of the TFs knockout effect on the target genes expression. Processings are carried out for each target gene successively. Indeed, the single target gene processing led to results comparable to the four target genes processing but much quicker. We choose the maximum number of changepoints $\bar{k} = 4$ and the maximal number of predictors within phases $\bar{s} = 4$. The number of iterations was 10 000, which was shown to be sufficient (results were similar to those obtained with 15 000 or 20 000 iterations). Each processing required around 2 minutes.

We carried out model selection based on the maximum of the posterior model probability $\hat{\mathbf{p}}(k|y)$ which was shown, through an extensive simulation study, to perform better than other classical criteria by Andrieu and Doucet [AD99]. Then the changepoints position is the maximum of $\hat{\mathbf{p}}(\xi|\hat{k}, y)$ the estimated posterior distribution of ξ given \hat{k} changepoints. Model structures within phases was selected according to conditional maxima of estimated posterior distributions as well. In short, the selected network for the four target genes are defined by parameters $(\hat{k}, \hat{\xi}, \hat{s}, \hat{\tau})$ where,

$$\hat{k} = \underset{0 \leq k \leq \bar{k}}{\operatorname{argmax}} \hat{\mathbf{p}}(k|y), \quad (4.34)$$

$$\hat{\xi} = \underset{\xi \subseteq N^p}{\operatorname{argmax}} \hat{\mathbf{p}}(\xi|\hat{k}, y), \quad (4.35)$$

and for all phase h of all gene i ,

$$\hat{s}_h^i = \underset{0 \leq s_h^i \leq \bar{s}}{\operatorname{argmax}} \hat{\mathbf{p}}(s_h^i|\hat{k}, \hat{\xi}, y), \quad (4.36)$$

$$\hat{\tau}_h^i = \underset{\tau_h^i \in Q}{\operatorname{argmax}} \hat{\mathbf{p}}(\tau_h^i|\hat{k}, \hat{\xi}, \hat{s}, y). \quad (4.37)$$

The detailed results obtained with our 2-steps reversible jump MCMC inference method for the 4 target genes FLR1, GTT2, TPO1, SNG1 appear in Appendix (subsections 4.5.3 to 4.5.6). On the second page of each subsection, the top left hand corner graph displays the change of the number of changepoints (between 0 and $\bar{k} = 4$) throughout the 5000 iterations. The top right hand corner graph displays the derived estimation of the posterior density $\hat{\mathbf{p}}(k|y)$. Then a table shows the estimated posterior density $\hat{\mathbf{p}}(\xi|y)$ by decreasing order. The two last figures expose the estimated posterior distributions $\hat{\mathbf{p}}(\tau_h^i|\hat{\xi}, y)$ for the subset of factor variables given the first $\hat{\xi}$ and the second $\hat{\xi}'$ most probable changepoint structure. To display these estimated posterior distributions, we number the 2^q possible model structures by the sum over the number attributed to each TF included in the model (see Table 4.2). For example, the model defined by the two predictors YAP1 and YRR1 is numbered $8 + 1 = 9$.

TF	YAP1	PDR1	PDR3	YRR1
Number	8	4	2	1

Table 4.2: Reference numbers attributed to each TF.

Results

The inferred chronological networks describing the effect of each of the four TF deletions on target gene coding for FLR1, GTT2, TPO1 and SNG1, are respectively exposed in Figures 4.11 to 4.14. The dependency structure is represented for each inferred phase successively. This points out the chronology of the detected interaction relationships. The top nodes represents the four TFs: YAP1, PDR1, PDR3 and YRR1 (left to right). The bottom nodes are for the studied target genes and its two nearest neighbors included into the analysis. When drawn in dotted line, the edges describe the second most probable network structure within a phase.

The inferred network describing the effect of each of the four TF deletions on target gene coding for FLR1 is divided into 3 phases: the first time point (+30 sec), the second time point (+2 min) and the three last time points (+4 to 20 min) after benomyl addition. In the second

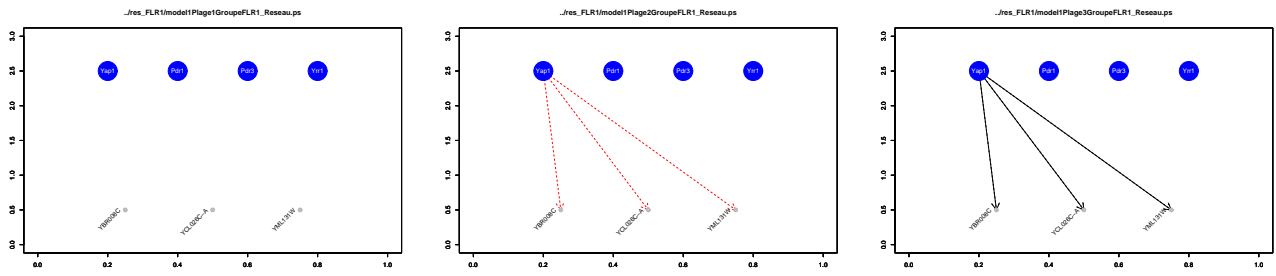


Figure 4.11: FLR1 inferred regulation network: 3 phases, that is the first time point (+30 sec), the second time point (+2 min) and the three last time points (+4 to 20 min) after benomyl addition. This points out FLR1 as an early target of YAP1.

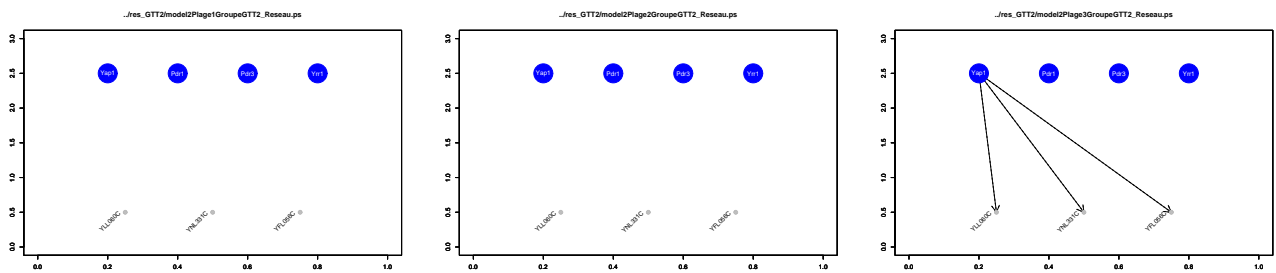


Figure 4.12: GTT2 inferred regulation network: 3 phases, that is (+30 sec), (+2 min) and (+4 to 20 min) after benomyl addition. GTT2 appears as latter target of YAP1.

phase (+2 to 4 min), the most probable network is the empty network (without edge). We represent in dotted line the second most probable network because its posterior probability is much higher than the other ones (see second graph of Figure 4.19 in Appendix). At time point (+2min), one out of the two nearest neighbor genes has a very low expression level. This may explain the “hesitation” observed at time point (+2min): it makes it difficult to give a decision on the effect of YAP1 deletion at this time point. As a consequence a changepoint is created and two different structures are pointed out for the second phase: the empty model and the model with an effect of YAP1. Except for the second one, the decision is obvious for all other time points. The difficulty met with time point 2 is notably due to the lack of repeated measurements. Nevertheless, FLR1 still appear as an early target of YAP1.

The inferred network for GTT2 has 3 phases two: the first time point (+30 sec), the second time point (+2 min) and the three last time points (+4 to 20 min). The most probable dependency structures of the two first phases are identical but the error variance differs (see Appendix 4.5.7, Figure 4.36). This explains the birth of the changepoint at (+30 sec). From these results, GTT2 appears as a later target of YAP1. When YAP1 is knocked out, the alteration of GTT2 gene expression only starts 4 minutes after benomyl addition.

TPO1 inferred regulation network has only one single phase. This structure is from a distance the most probable and TPO1 is strongly affected by PDR1 deletion during the whole time of the experiment.

The inferred network for SNG1 contains 3 phases: the first time point (+30 sec), the second time point (+2 min) and the three last time points (+4 to 20 min). As for GTT2, the existence of the changepoint at time point (+30 sec) is explained by the increase of the the error variance in

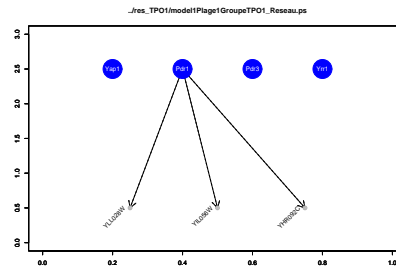


Figure 4.13: TPO1 inferred regulation network: 1 single phase. TPO1 is strongly affected by PDR1 deletion during the whole time of the experiment.

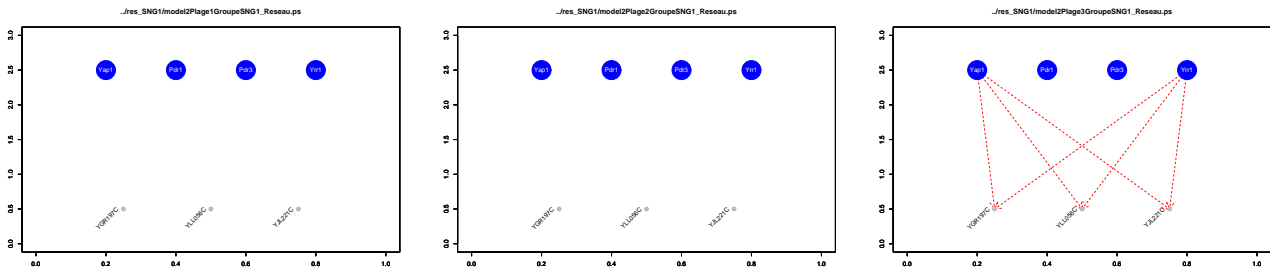


Figure 4.14: SNG1 inferred regulation network: 3 phases, that is (+30 sec), (+2 min) and (+4 to 20 min) after benomyl addition. The procedure does not give a straight results because the data does not contain expression profiles very similar to SNG1 (see Appendix 4.5.6).

the second phase (see Appendix 4.5.7, Figure 4.37). In the third phase (from +4 to +20 min), the network is represented in red because it is the second most probable. The most probable is the full network (with all edges) which is not a biologically acceptable result. The second most probable is the model with a joint effect of YAP1 and YRR1 deletion represented in dotted line in Figure 4.14. Actually there are three model structures with a posterior probability much higher than others (the third most probable is the empty network, see Appendix 4.35). This comes from the fact that the data does not contain expression profiles very similar to SNG1. Indeed, one out of the two nearest expression profiles differ a lot from SNG1 expression profile for PDR1 and PDR3 deleted strains (see Figure 4.31 in Appendix). As the expression of this gene is affected by PDR1 and PDR3 deletions, the full network is pointed as the most probable. Thus the obtained result is coherent with the data but the analysis of these three genes can not be carried out globally.

4.4.3 Conclusion and future work

The novelty of the approach developed in this chapter is first to allow the inference of a time-dependent network, that is to allow the inference of both the changepoints position and the dependency structure within phases, and second to do it simultaneously.

The first results obtained for genes implicated in the response to benomyl by *S. cerevisiae* are very promising, especially if we can hope to get repeated measurements data later. As we do not dispose of repeated measurements for the benomyl answer analysis yet, we proposed to carry out global analysis on 3-gene clusters, made up of a target gene and its two nearest neighbors. The approach is efficient for the genes having “close” neighbors. Indeed, this study notably allows to point out the chronological effect of YAP1 deletion: FLR1 is pointed out as an earlier target than

GTT2. Even for TPO1 whose PDR1 deleted strain expression profiles does not much differ from the other strain expression profiles, the effect of PDR1 deletion is detected.

Of course, we cannot decide “exceptional” genes’ case, that is genes to which no other gene expression profile is similar, unless we had repeated measurements. This emphasizes the need for determining in which manner it is acceptable to cluster genes and to carry out an common analysis for each cluster. Thus we are carrying out a deep analysis which aims at setting conditions allowing such a global analysis. At least the procedure does not give a straight result when the data are not “homogeneous enough” (like for SNG1). This protects from an abusive use of clustering.

The creation of a changepoint when variance differs from one time point to the next one is also a point that need to be discussed. On the one hand, when this happens, it is a fact that the variances differ and we are interested in knowing it. On the other hand, this increases a lot the dimension of the model. Thus question is do we want to impose the same variance over the whole experiment? This is a point to be enlightened. We are working at evaluating the pertinence and efficiency of a single variance model.

Once a stand is taken towards these two points, this approach for time-dependent network inference will allow to carry out a comprehensive analysis over all target genes, at least for the genes whose available data allows such a study.

4.5 Appendix

4.5.1 Nomenclature

n	number of time points,
p	number of target genes,
q	number of factor genes,
$N = \{1, \dots, n\}$	set of the time points,
$P = \{1, \dots, p\}$	set of the target genes,
$Q = \{1, \dots, q\}$	set of the factor genes,
Y_t^i	random variable i observed at time t ($1 \leq t \leq N$),
X_t^j	predictor random variable j observed at time t ($1 \leq t \leq N$),
m^i	number of repeated measurements for each time point of gene i ,
y_{tl}^i	l^{th} observed value for Y_t^i ($1 \leq l \leq m^i$),
x_{tl}^j	l^{th} observed value for X_t^j ,
$y = \{y_{tl}^i; \forall i, t, l\}$	target genes observation data
$x = \{x_{tl}^j; \forall j, t, l\}$	factor genes observation data
\bar{k}	maximal number of changepoints,
k^i	number of changepoints for gene i ($0 \leq k^i \leq \bar{k}$),
$k = \sum_{i=1}^p k^i$	total number of changepoints,
$\xi^i = (\xi_0^i, \xi_1^i, \dots, \xi_{k^i+1}^i)$	changepoints position vector for target gene i , with $\xi_{h-1}^i < \xi_h^i$ and $\xi_0^i = 1$ and $\xi_{k^i+1}^i = n + 1$,
$\xi = (\xi^i; 1 \leq i \leq p)$	full changepoints position vector
\bar{s}	maximal number of predictors in each regression model
s_h^i	number of explicative variables for gene i in phase h ($0 \leq s_h^i \leq \bar{s}$),
τ_h^i	set of predictor variables for gene i in phase h ($ \tau_h^i = s_h^i$),
$\theta_h^i = ((a_h^{ij})_{j \in 0..q}, \sigma_h^i)$	parameters defining the regression model in phase h for gene i .
Probability distributions	
Uniform	$\mathcal{U}_A \quad [\int_A d\mathbf{z}]^{-1} \mathbf{1}_A(\mathbf{z})$
Gaussian	$\mathcal{N}(\boldsymbol{\mu}, \Sigma) \quad 2\pi\Sigma ^{-1/2} \exp(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^t \Sigma^{-1}(\mathbf{z} - \boldsymbol{\mu}))$
Gamma	$\mathcal{G}a(\alpha, \beta) \quad (\beta^\alpha / \Gamma(\alpha)) z^{\alpha-1} \exp(-\beta z) \mathbf{1}_{[0, +\infty[}(z)$
Inverse Gamma	$\mathcal{I}G(\alpha, \beta) \quad (\beta^\alpha / \Gamma(\alpha)) z^{-\alpha-1} \exp(-\beta/z) \mathbf{1}_{[0, +\infty[}(z)$

Figure 4.15: Nomenclature.

4.5.2 Birth of a new predictor for phase h of gene i acceptance probability

$$r_{s_h^i, s_h^i+1}(\tau_h^i, \tau_h^{i+}) = \frac{\Pi_{s_h^i+1}(s_h^i+1, \tau_h^{i+} | y_h^i) d_{s_h^i+1} q(j^* | \tau_h^{i+})}{\Pi_{s_h^i}(s_h^i, \tau_h^i | y_h^i) b_{s_h^i} q(j^* | \tau_h^i)}, \quad (4.38)$$

with,

$$\begin{aligned} \Pi_{s_h^i}(s_h^i, \tau_h^i | y_h^i) &\propto (\gamma_0 + (y_h^i)^t P_{\tau_h^i} y_h^i)^{-(m^i(\xi_h^i - \xi_{h-1}^i) + v_0)/2} \frac{(\lambda/\sqrt{1+\delta^2})^{s_h^i}}{s_h^i! \binom{q}{s_h^i}}, \\ q(j^* | \tau_h^{i+}) &= \frac{1}{s_h^i+1}, \\ q(j^* | \tau_h^i) &= \frac{1}{q-s_h^i}, \end{aligned}$$

where, for any subset τ_h^i of regressors, denoting by I the identity matrix of size $m^i(\xi_h^i - \xi_{h-1}^i)$, $P_{\tau_h^i}$ is a projection matrix defined as follows,

$$P_{\tau_h^i} = I - \frac{\delta^2}{\delta^2 + 1} D_{\tau_h^i}(x) \left(D_{\tau_h^i}^t(x) D_{\tau_h^i}(x) \right)^{-1} D_{\tau_h^i}^t(x).$$

After simplifications,

$$r_{s_h^i, s_{h+1}^i}(\tau_h^i, \tau_h^{i+}) = \frac{1}{\sqrt{1 + \delta^2}} \left(\frac{\gamma_0 + (y_h^i)^t P_{\tau_h^i} y_h^i}{\gamma_0 + (y_h^i)^t P_{\tau_h^{i+}} y_h^i} \right)^{(m^i(\xi_h^i - \xi_{h-1}^i) + \nu_0)/2}.$$

4.5.3 FLR1: YAP1 early target.

Collected data

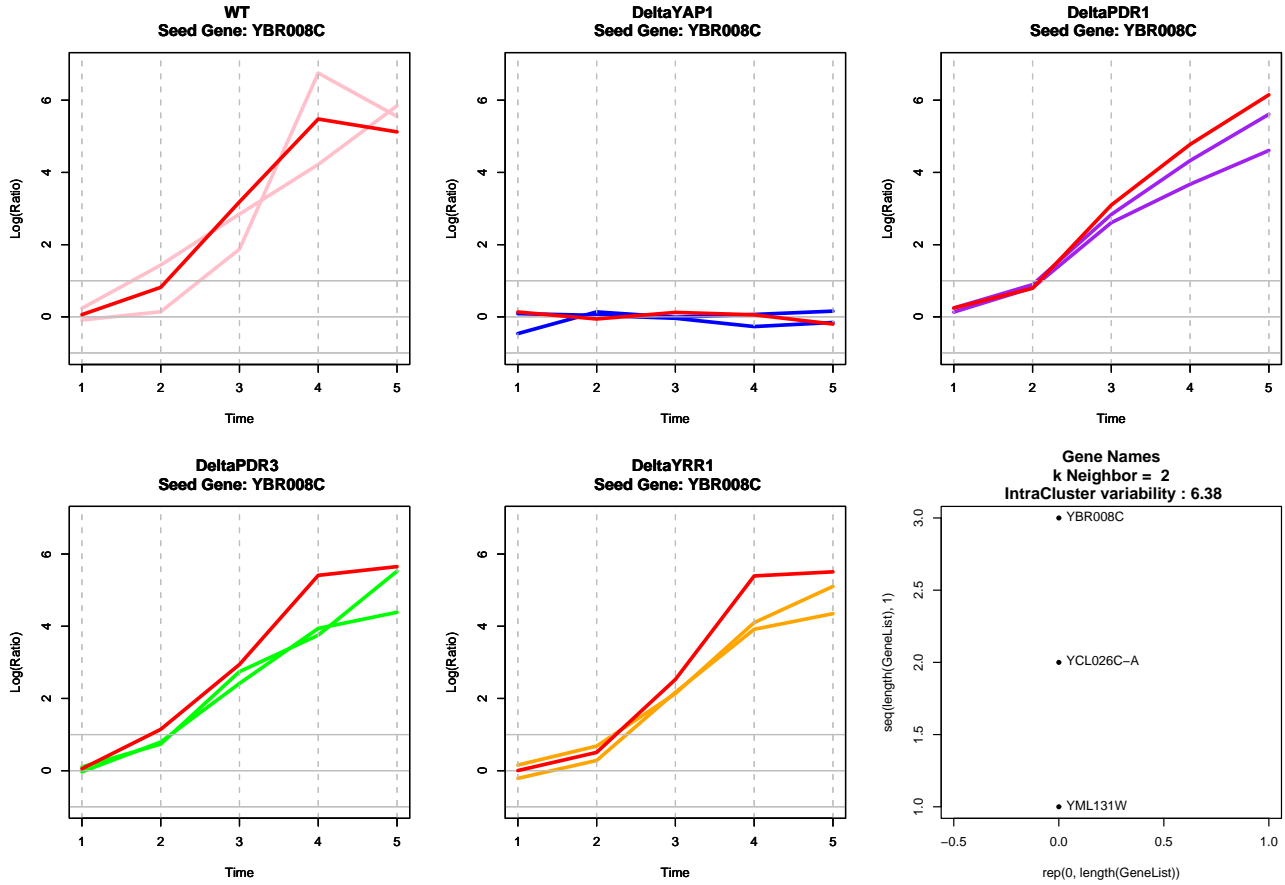


Figure 4.16: Expression profiles of gene coding for protein FLR1 (red plotted) and its 2 nearest neighbors for the 5 strains under study (from left to right: wild type, and successively YAP1, PDR1, PDR3 and YRR1 deleted).

Inferred network

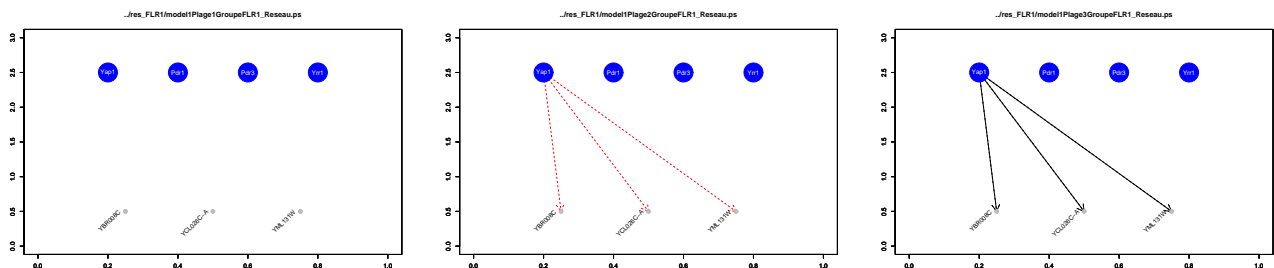


Figure 4.17: Most probable network: 3 phases, that is the first time point (+30 sec), the second time point (+2 min) and the three last time points (+4 to 20 min) after benomyl addition. This points out FLR1 as an early target of YAP1.

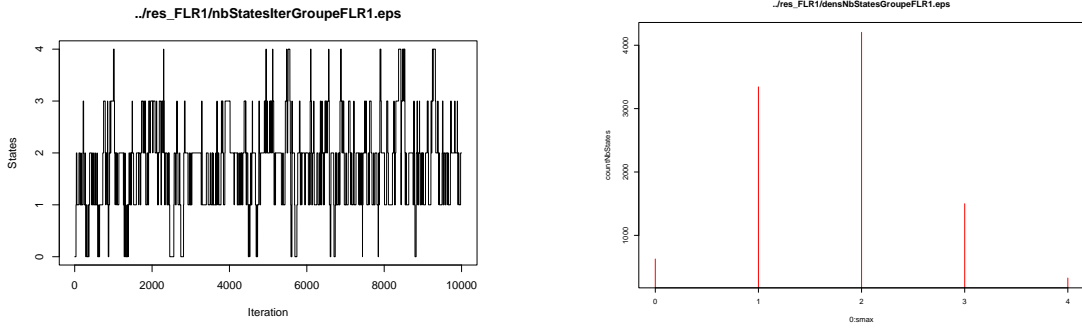


Figure 4.18: Left: Number of changepoints throughout the iterations for FLR1. Right: estimation of the posterior distributions $\mathbf{p}(k|y)$.

$\hat{\mathbf{p}}(\xi y)$	ξ
0.2962	(1, 2, 3, 6)
0.2638	(1, 2, 6)
0.0627	(1, 6)
0.0581	(1, 2, 3, 4, 6)
0.0396	(1, 2, 5, 6)
0.038	(1, 3, 4, 5, 6)
0.0327	(1, 2, 3, 4, 5, 6)
0.0322	(1, 3, 4, 6)

Table 4.3: Estimated posterior density $\hat{\mathbf{p}}(\xi|y)$.

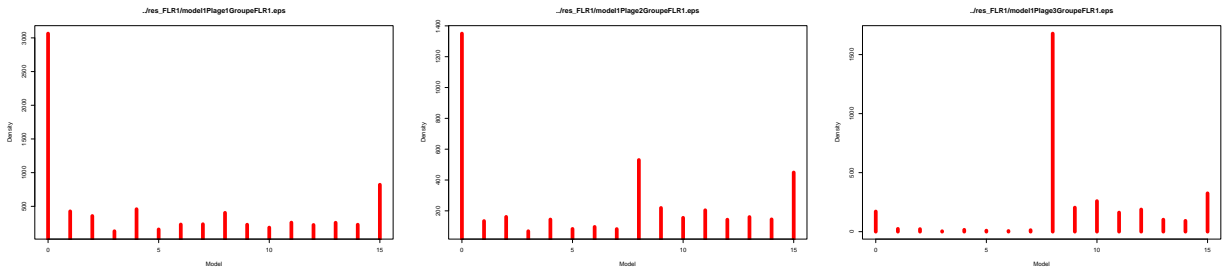


Figure 4.19: Estimation of the posterior distributions of $\mathbf{p}(\tau_h^i|\hat{\xi}, y)$ for each phase delimited by the inferred changepoint $\hat{\xi}$ as defined by equation (4.35).

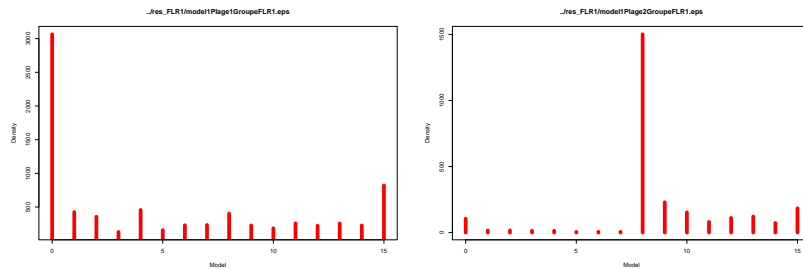


Figure 4.20: Estimation of the posterior distributions of $\mathbf{p}(\tau_h^i|\hat{\xi}', y)$ for each phase delimited by the most probable changepoints vector $\hat{\xi}'$ whose number of changepoint k' is the second most probable.

4.5.4 GTT2: YAP1 latter target.

Collected data

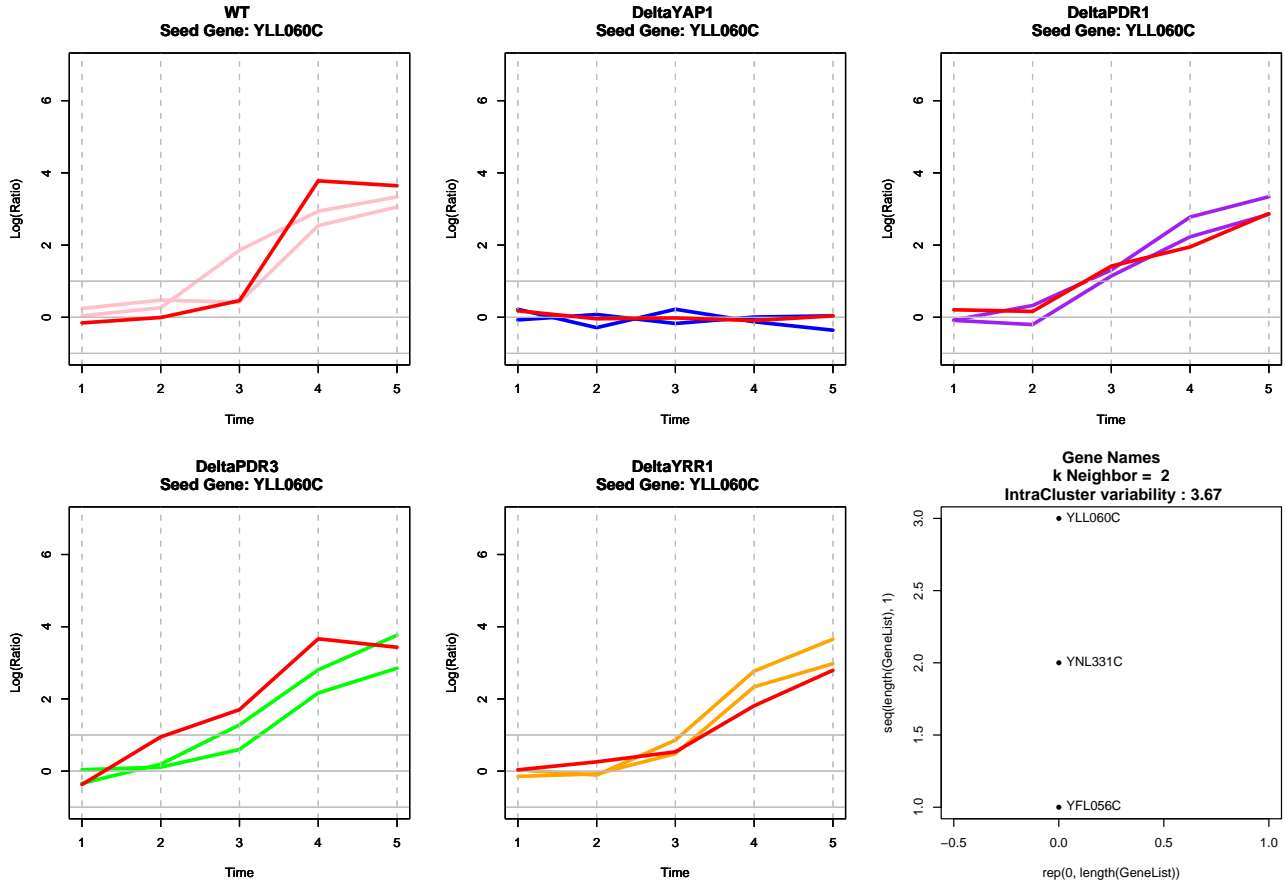


Figure 4.21: Expression profiles of gene coding for protein GTT2 (red plotted) and its 2 nearest neighbors for the 5 strains under study (from left to right: wild type, and successively YAP1, PDR1, PDR3 and YRR1 deleted).

Inferred network

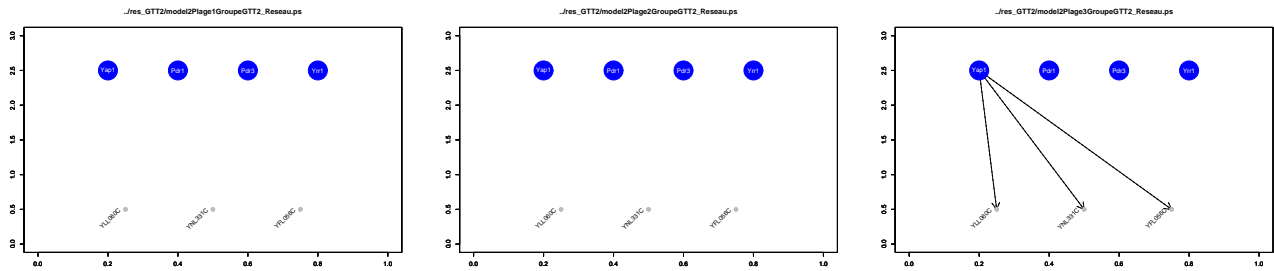


Figure 4.22: Most probable network: 3 phases, that is (+30 sec), (+2 min) and (+4 to 20 min) after benomyli addition.

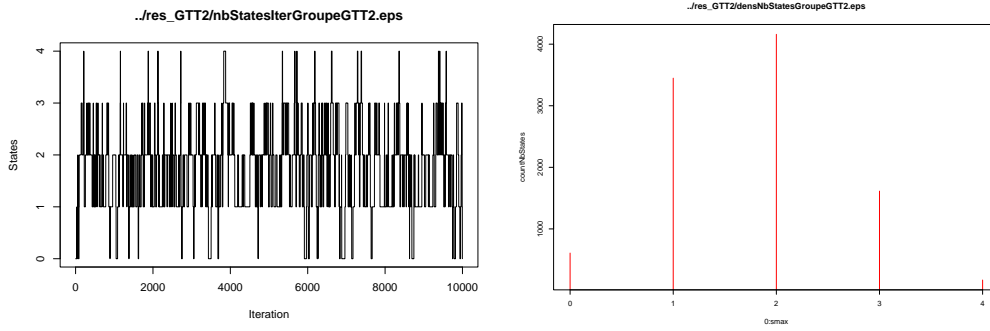


Figure 4.23: Left: number of changepoints throughout the iterations for GTT2. Right: estimation of the posterior distributions $\mathbf{p}(k|y)$.

$\hat{\mathbf{p}}(\xi y)$	ξ
0.2461	(1, 2, 6)
0.1597	(1, 2, 3, 6)
0.0777	(1, 2, 5, 6)
0.0612	(1, 3, 4, 6)
0.0608	(1, 6)
0.0545	(1, 5, 6)
0.0526	(1, 4, 5, 6)
0.0517	(1, 2, 3, 4, 6)

Table 4.4: Estimated posterior density $\hat{\mathbf{p}}(\xi|y)$.

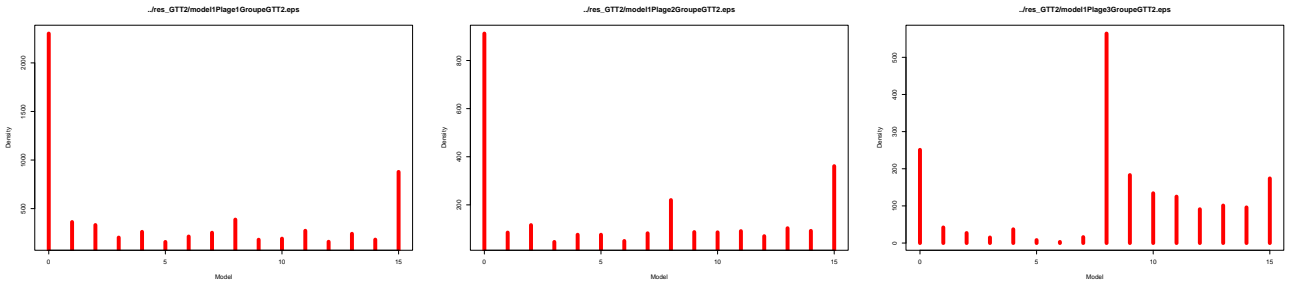


Figure 4.24: Estimation of the posterior distributions of $\mathbf{p}(\tau_h^i|\hat{\xi}, y)$ for each phase delimited by the inferred changepoint $\hat{\xi}$ as defined by equation (4.35).

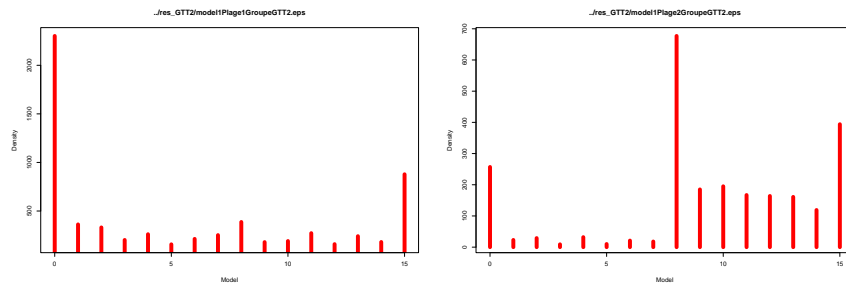


Figure 4.25: Estimation of the posterior distributions of $\mathbf{p}(\tau_h^i|\hat{\xi}', y)$ for each phase delimited by the most probable changepoints vector $\hat{\xi}'$ whose number of changepoint k' is the second most probable.

4.5.5 TPO1: PDR1 target.

Collected data

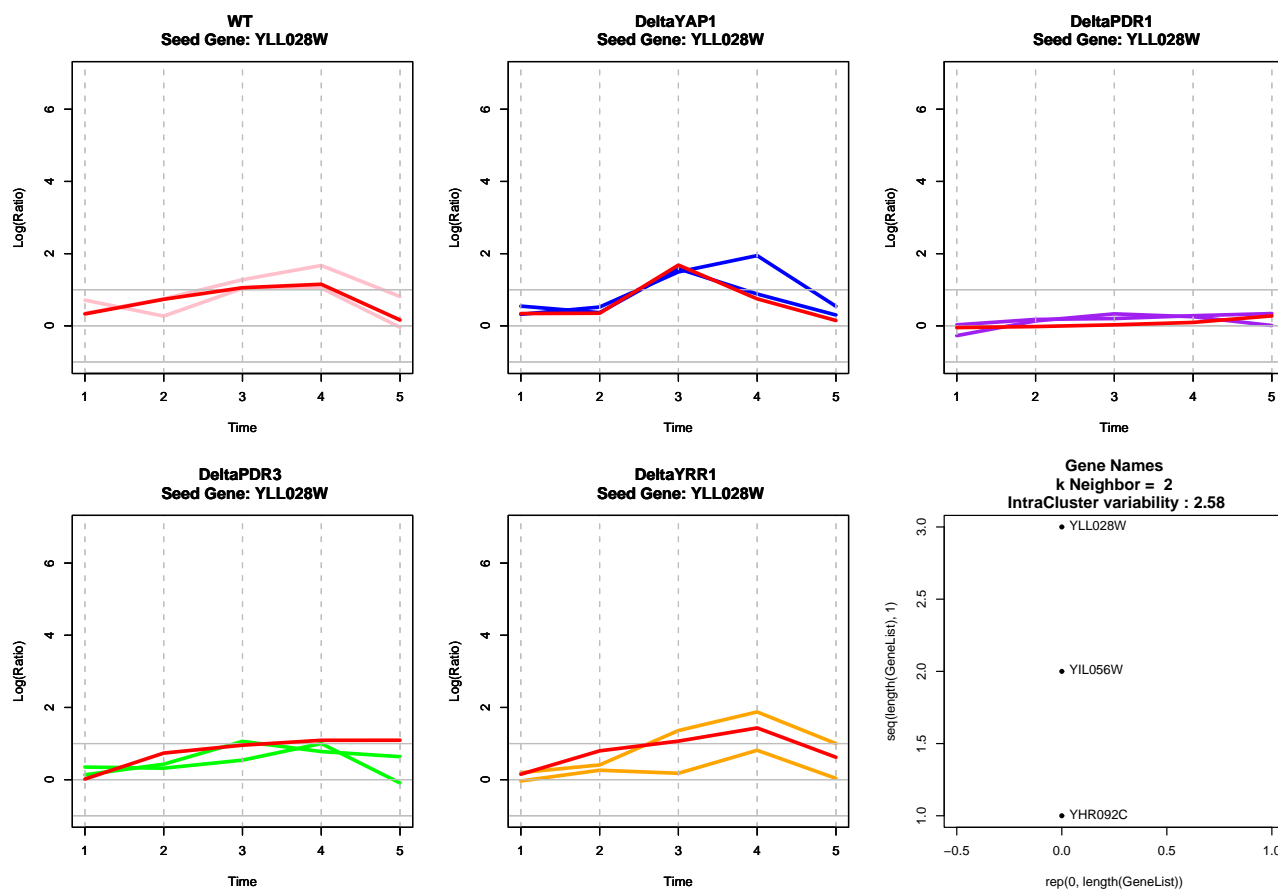


Figure 4.26: Expression profiles of gene coding for protein TPO1 (red plotted) and its 2 nearest neighbors for the 5 strains under study (from left to right: wild type, and successively YAP1, PDR1, PDR3 and YRR1 deleted).

Inferred network

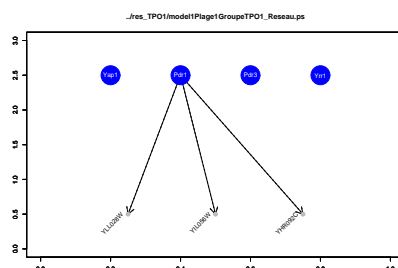


Figure 4.27: Most probable network: 1 single phase. TPO1 is strongly affected by PDR1 deletion during the whole time of the experiment.

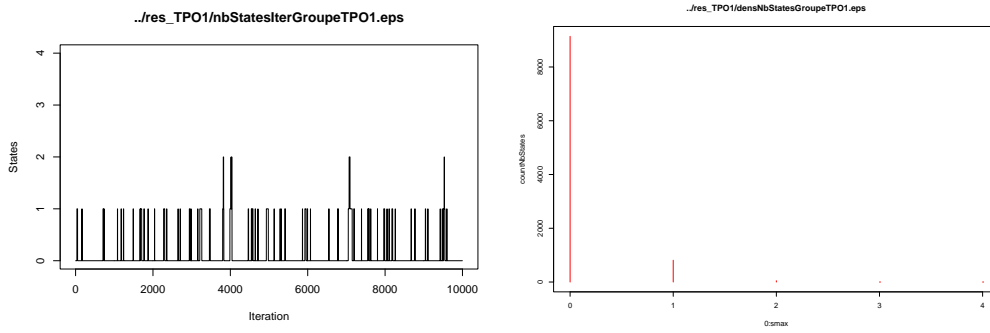


Figure 4.28: Left: number of changepoints throughout the iterations for TPO1. Right: estimation of the posterior distributions $\mathbf{p}(k|y)$.

$\hat{\mathbf{p}}(\xi y)$	ξ
0.9141	(1, 6)
0.0302	(1, 2, 6)
0.0249	(1, 5, 6)
0.0158	(1, 3, 6)
0.0102	(1, 4, 6)
0.003	(1, 4, 5, 6)
0.0018	(1, 2, 3, 6)

Table 4.5: Estimated posterior density $\hat{\mathbf{p}}(\xi|y)$.

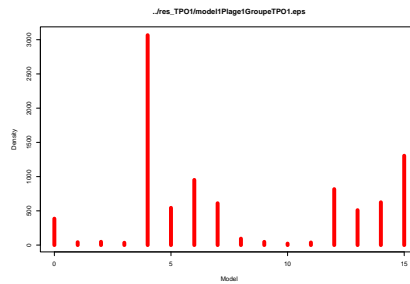


Figure 4.29: Estimation of the posterior distributions of $\mathbf{p}(\tau_h^i|\hat{\xi}, y)$ for each phase delimited by the inferred changepoint $\hat{\xi}$ as defined by equation (4.35).

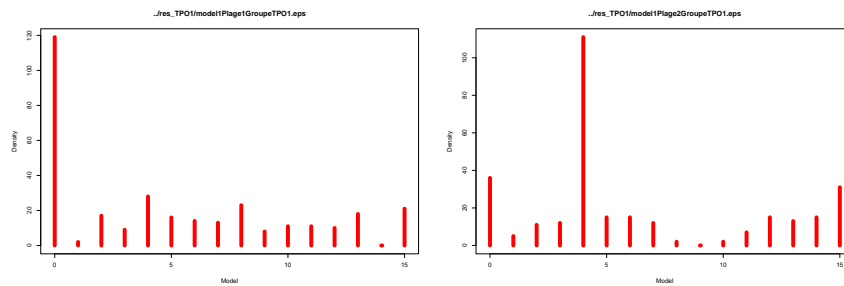


Figure 4.30: Estimation of the posterior distributions of $\mathbf{p}(\tau_h^i|\hat{\xi}', y)$ for each phase delimited by the most probable changepoints vector $\hat{\xi}'$ whose number of changepoint k' is the second most probable.

4.5.6 SNG1: YRR1 and YAP1 target.

Collected data

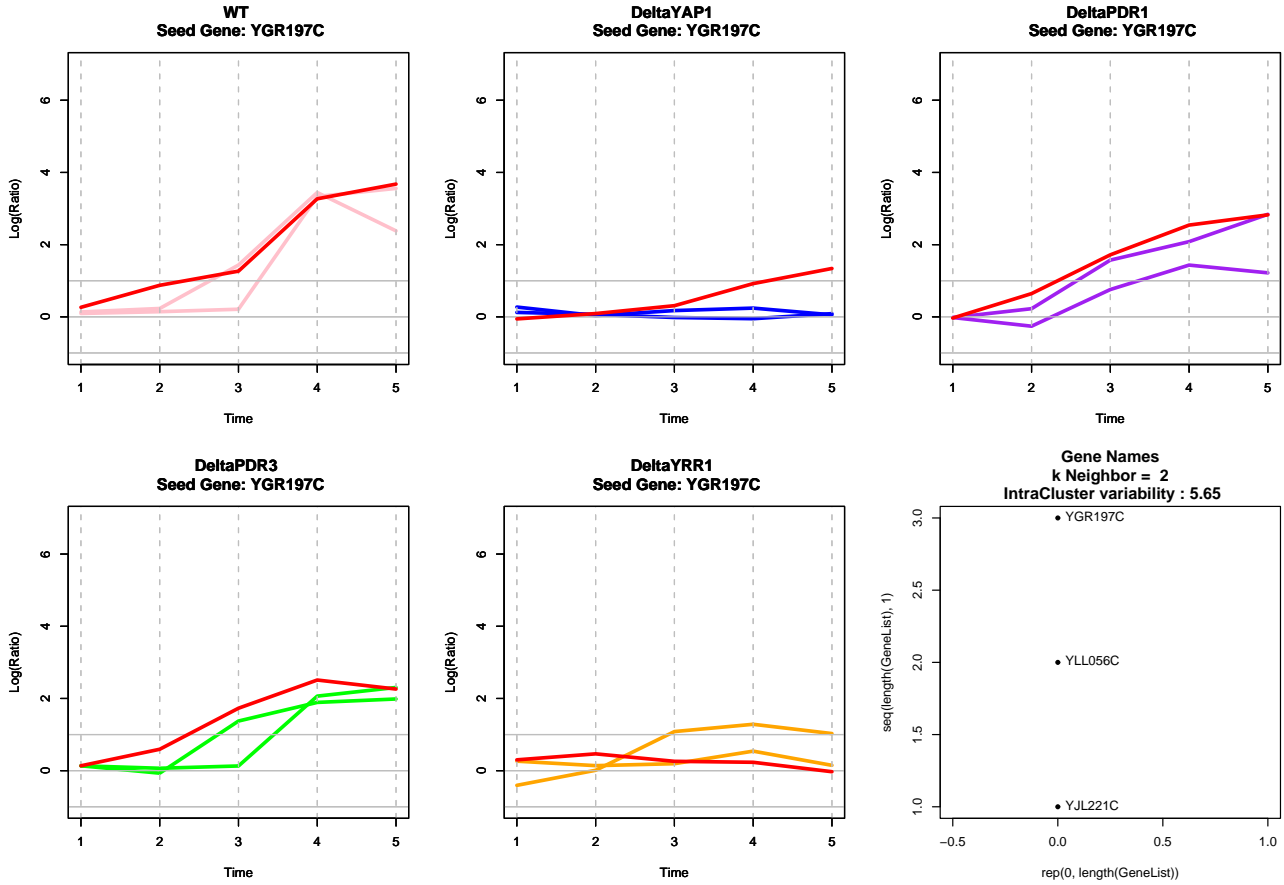


Figure 4.31: Expression profiles of gene coding for protein SNG1 (red plotted) and its 2 nearest neighbors for the 5 strains under study (from left to right: wild type, and successively YAP1, PDR1, PDR3 and YRR1 deleted).

Inferred network

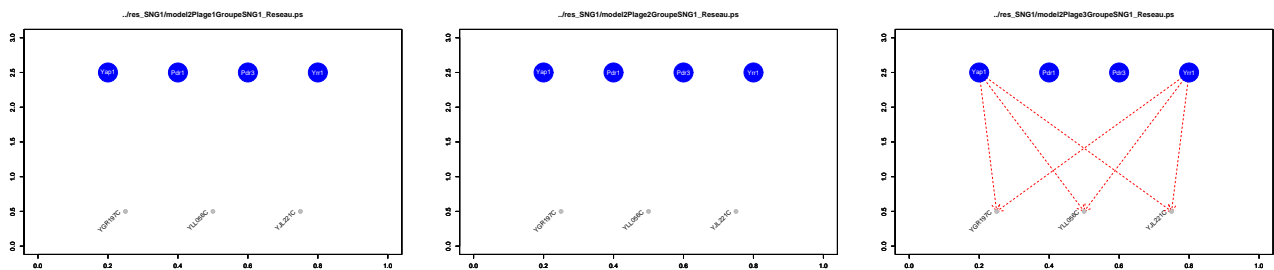


Figure 4.32: Most probable network: 3 phases, that is (+30 sec), (+2 min) and (+4 to 20 min) after benomyl addition. The procedure does not give a straight results because the data does not contain expression profiles very similar to SNG1 (see Figure 4.31).

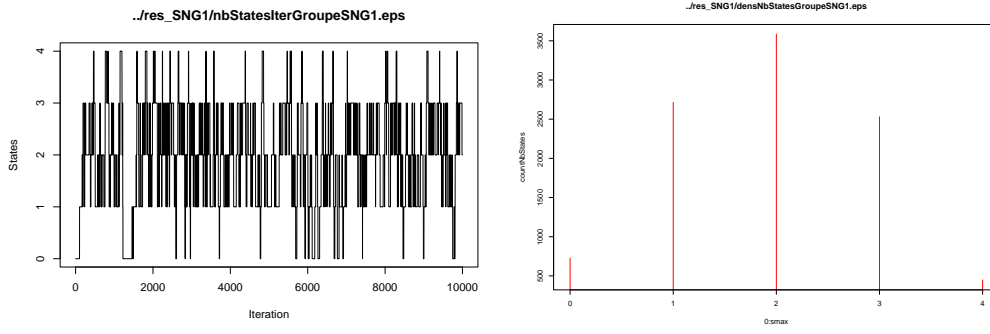


Figure 4.33: Left: number of changepoints throughout the iterations for SNG1. Right: estimation of the posterior distributions $\mathbf{p}(k|y)$.

$\hat{\mathbf{p}}(\xi y)$	ξ
0.2053	(1, 2, 6)
0.1287	(1, 2, 3, 6)
0.0919	(1, 2, 3, 4, 6)
0.0744	(1, 3, 4, 5, 6)
0.0725	(1, 6)
0.0641	(1, 2, 5, 6)
0.0641	(1, 3, 4, 6)
0.0451	(1, 2, 4, 5, 6)

Table 4.6: Estimated posterior density $\hat{\mathbf{p}}(\xi|y)$.

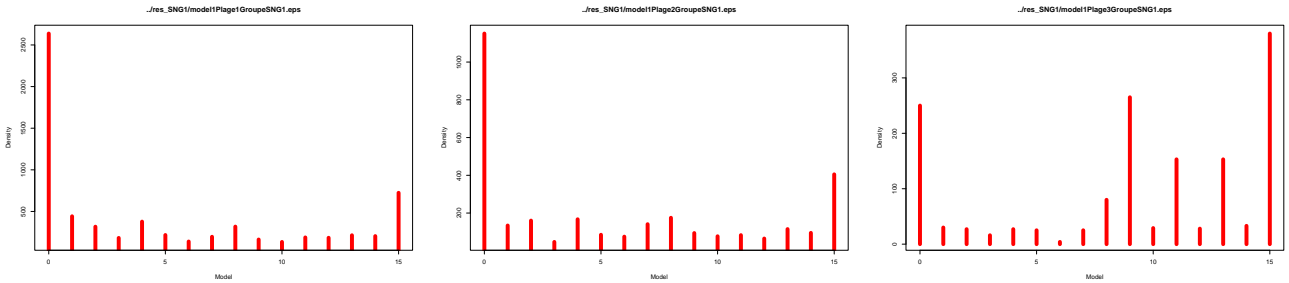


Figure 4.34: Estimation of the posterior distributions of $\mathbf{p}(\tau_h^i|\hat{\xi}, y)$ for each phase delimited by the inferred changepoint $\hat{\xi}$ as defined by equation (4.35).

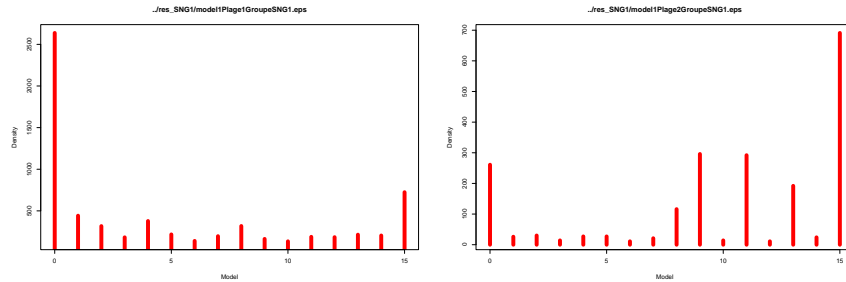


Figure 4.35: Estimation of the posterior distributions of $\mathbf{p}(\tau_h^i|\hat{\xi}', y)$ for each phase delimited by the most probable changepoints vector $\hat{\xi}'$ whose number of changepoint k' is the second most probable.

4.5.7 Estimated posterior distributions for the error variance of GTT2 and SNG1.

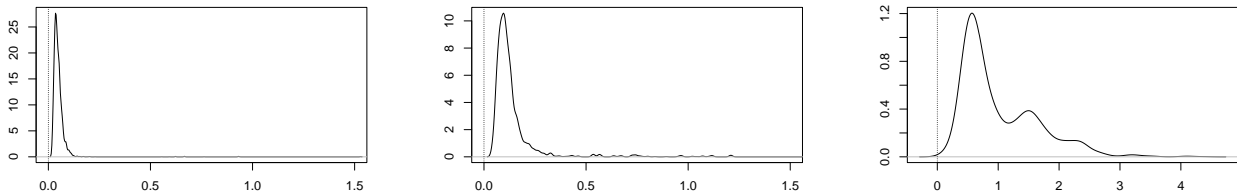


Figure 4.36: Estimation of the posterior distributions of $\mathbf{p}(\sigma_h^i | \hat{\xi}, y)$ for each phase delimited by the inferred changepoint $\hat{\xi}$ for GTT2.

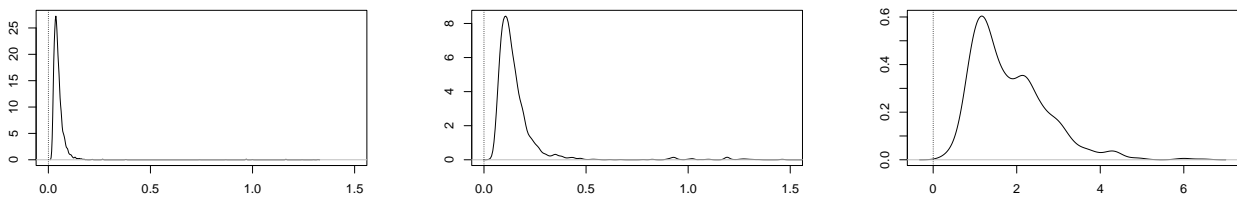


Figure 4.37: Estimation of the posterior distributions of $\mathbf{p}(\sigma_h^i | \hat{\xi}, y)$ for each phase delimited by the inferred changepoint $\hat{\xi}$ for SNG1.

Appendix A

Additional Files: R package 'G1DBN' reference manual

Description des fonctions implémentées dans le package G1DBN,

- inferG1,
- GfromG1,
- edges,
- simulAR1,
- arth800line,

pour l'inférence de réseaux bayésiens dynamiques à partir d'indépendances partielles d'ordre 1 exposée dans le chapitre 3.

inferG1*First order dependence graph G(1) inference*

Description

This function infers the score $S1$ of each potential edge of a Dynamic Bayesian Network (defined by full order dependence DAG G) by considering 1st order dependencies. The smallest score points out the most significant edge for the 1st order dependence DAG $G(1)$. `inferG1` is the 1st step of the estimation procedure described in the references.

Usage

```
out<-inferG1(data,ls=T,huber=F,tukey=F,kendall=F,predictor=NULL,
target=NULL)
```

Arguments

data	a matrix with n rows (=time points) and p columns (=genes) containing the gene expression time series.
ls	boolean, TRUE to obtain a score matrix computed by using Least Square estimator, default=TRUE.
huber	boolean, TRUE to obtain a score matrix computed by using Huber estimator, default=FALSE.
tukey	boolean, TRUE to obtain a score matrix computed by using Tukey bisquare (or biweight) estimator, default=FALSE.
predictor	To be specified if the possible predictor genes should be reduced to a subset of d predictor genes: an array included in $[1,p]$ defining the position of the d predictor genes in the data matrix, default=NULL.
target	To be specified if the possible target genes should be reduced to a subset of r target genes: an array included in $[1,p]$ defining the position of the r target genes in the data matrix, default=NULL.

Value

A list with `out$ls` a matrix with $\min(r,p)$ columns (=target genes) and $\min(d,p)$ rows (=predictor genes) containing the scores $S1$ obtained with least square estimator (`out$ls[i,j]` is the score for the edge $j \rightarrow i$), `out$huber` a matrix containing scores $S1$ obtained with Huber estimator, `out$tukey` a matrix containing scores $S1$ obtained with Tukey bisquare (or biweight) estimator.

Note

For a large number of target genes, it may be interesting to parallel run the procedure `inferG1` for each target gene.

Author(s)

Lebre Sophie (<http://stat.genopole.cnrs.fr/~slebre>).

References

Lebre, S. 2007. Inferring Dynamic Bayesian Networks with low order dependencies. Preprint available at <http://hal.archives-ouvertes.fr/hal-00142109>.

See Also

GfromG1, edges.

Examples

```
# load G1DBN Library
library(G1DBN)

data(arth800line)
data<-arth800line
id<-c(60, 141, 260, 333, 365, 424, 441, 512, 521, 578, 789, 799)

# compute score S1
pmaxG1<-inferG1(data,ls=TRUE,tukey=FALSE,huber=FALSE,predictor=id,target=id)
round(pmaxG1$ls,2)

resG1<-edges(score=pmaxG1$ls,targetNames=id,predNames=id,validMat=NULL,roc=FALSE,
threshold=0.001,nb=NULL,prec=6)
resG1$nameslist

# compute score S2 from S1
GwithLS<-GfromG1(pmaxG1$ls,data,method='ls',alpha1=0.1,alpha2=0.01,predictor=NULL,
target=NULL)
GwithLS

resG<-edges(score=GwithLS$S2,targetNames=id,predNames=id,validMat=NULL,roc=FALSE,
threshold=0.001,nb=NULL,prec=6)
resG$nameslist
```

GfromG1

Full order dependence DAG G inference

Description

This function infers the scores of each potential edge of a Dynamic Bayesian Network (defined by full order dependence DAG G) from a score matrix S1 (obtained with function inferG1) which describes the score of the edges for the 1st order dependence DAG G(1). This is the second step of the inference procedure described in the references: 1st step inferG1 allows to reduce the number of potential edges, GfromG1 performs the last step selection. The smallest score points out the most significant edge.

Usage

```
out<-GfromG1(S1,data,method='ls',alpha1,alpha2=1,predictor=NULL,
target=NULL)
```

Arguments

S1	a matrix with r rows (=target genes) and d columns (=predictor genes) containing score S1 (maximal p-value) obtained with function inferG1.
data	the matrix with n rows (=time points) and p columns (=genes) containing the corresponding gene expression time series.
method	currently M estimation with either LS, Tukey bisquare or Huber estimator, c('ls', 'tukey', 'huber'), default='ls'.
alpha1	Step 1 threshold for edge selection in $G(1)$.
alpha2	Step 2 threshold for edge selection in G , default=1.
predictor	To be specified if the possible predictor genes should be restricted to a subset of d predictor genes: an array included in $[1,p]$ defining the position of the d predictor genes in the data matrix, default=NULL.
target	To be specified if the possible target genes should be reduced to a subset of r target genes: an array included in $[1,p]$ defining the position of the r target genes in the data matrix, default=NULL.

Value

A list with `out$S2` a matrix (r rows, d columns) containing the scores obtained with the chosen M estimator, `out$maxInG1` the maximal number of parent in DAG $G(1)$ for the chosen threshold `alpha1`, `out$NbG1` the number of edges in 1st order dependence DAG $G(1)$, `out$NbG` the number of edges in full order dependence DAG G .

Author(s)

Lebre Sophie (<http://stat.genopole.cnrs.fr/~slebre>).

References

Lebre, S. 2007. Inferring Dynamic Bayesian Networks with low order dependencies. Preprint available at <http://hal.archives-ouvertes.fr/hal-00142109>.

See Also

inferG1, edges.

Examples

```
# load G1DBN Library
library(G1DBN)

data(arth800line)
data<-arth800line
id<-c(60, 141, 260, 333, 365, 424, 441, 512, 521, 578, 789, 799)

# compute score S1
pmaxG1<-inferG1(data,ls=TRUE,tukey=FALSE,huber=FALSE,predictor=id,target=id)
round(pmaxG1$ls,2)

# compute score S2 from S1
```

```
GwithLS<-GfromG1(pmaxG1$ls, data, method='ls', alpha1=0.1, alpha2=0.01,
predictor=NULL, target=NULL)
GwithLS

resG<-edges(score=GwithLS$S2,targetNames=id,predNames=id,validMat=NULL,roc=FALSE,
threshold=0.001,nb=NULL,prec=6)
resG$nameslist

# As the number of genes is reduced to 10 here, this results slightly differ
# from the results obtained in the Preprint cited in References.
```

edges

Edges listing and evaluation

Description

This function allows to order the edges by decreasing likelihood. ROC curves can also be computed if a validation matrix is specified.

Usage

```
out<-edges(score,predNames=NULL,targetNames=NULL,validMat=NULL,
roc=FALSE,threshold=1, nb=NULL, prec=3)
```

Arguments

score	matrix with r columns (=target genes) and d rows (=predictor genes) containing the scores resulting from an estimation procedure (either inferG1 or GfromG1).
predNames	An optional array (d) giving a list of names for the predictor genes, default=NULL.
targetNames	An optional array (r) giving a list of names for the target genes, default=NULL.
validMat	An optional matrix specifying the validated edges (1 if an edge is validated, 0 otherwise).
roc	Boolean, TRUE wether ROC curves should be computed, default=NULL. This option require to specify a validation matrix validMat.
threshold	An optional real setting the maximal value for edge selection, default=1.
nb	An optional integer setting the maximal number of selected edges, default=NULL.
prec	An optional integer setting the number of decimal places for score display, default=3.

Value

A list with `out$ref` and `out$names` matrices containing a list of edges ordered by decreasing likelihood (First column: parent, second column: child, third column: the corresponding score). If a validation matrix `validMat` is specified, a 4th column indicates wether the edge is validated (1 for a validated edge, 0 otherwise).

`out$ref` lists edges according to the index value of each variable and `out$names` uses the names from arrays `predNames` and `targetNames`.

`out$rocx` and `out$rocy` contain the coordinates for plotting ROC curves if a validation matrix is specified.

Author(s)

Lebre Sophie (<http://stat.genopole.cnrs.fr/~slebre>).

See Also

inferG1, GfromG1.

Examples

```
#generate AR(1) time series
AR<-simulAR1(p=10,n=50,edgeProp=0.02,minA=0.5,maxA=1.5,minB=0,maxB=1,minSig=0.1,maxSig=0.8)
AR

# compute score S1
pmaxG1<-inferG1(AR$data, ls=TRUE, tukey=FALSE, huber=FALSE, predictor=NULL, target=NULL)
pmaxG1$ls
# compute score S2 from S1
Gls<-GfromG1(pmaxG1$ls, AR$data, method='ls',alpha1=0.2,alpha2=1,predictor=NULL,
target=NULL)
Gls$score

### Results (The validation matrix is obtained from the simulation matrix AR$A).
resG1<-edges(score=pmaxG1$ls, targetNames=NULL, predNames=NULL,
validMat = (abs(AR$A)>0)*1, roc=TRUE, threshold=1, nb=NULL, prec=4)
resG<-edges(score=Gls$S2, targetNames=NULL, predNames=NULL,
validMat=(abs(AR$A)>0)*1, roc=TRUE, threshold=1, nb=NULL, prec=4)

# Edges list
resG1$list
resG$list

# ROC curve
plot(resG1$rocx/max(resG1$rocx),resG1$rocy/max(resG1$rocy), type="l",
xlab="False Positive", ylab="True Positive", main="ROC curve")
lines(resG$rocx/max(resG$rocx), resG$rocy/max(resG$rocy), col=2,lty=2)
leg=c("Step 1", "Step 2")
legend(0.8,0.2, leg, lty=c(1,2), col=c(1,2))
```

simulAR1

1st-order multivariate Auto-Regressive process generation

Description

This function generates multivariate time series according to the following first order Auto-Regressive process,

$$X(t) = A X(t-1) + B + e(t),$$

where matrix A has size $p \times p$ and arrays $X(t)$, B and $e(t)$ have length p. $e(t)$ follows a zero-centered multivariate gaussian distribution whose variance matrix S is diagonal. First, matrix A, array B

and diagonal of S are randomly generated. Each diagonal term $S[i,i]$ is uniformly generated from $U([\text{minSig}, \text{maxSig}])$. The elements of matrix A and array B are uniformly generated from $U([- \text{maxA}, - \text{minA}], [\text{minA}, \text{maxA}])$ and $U([\text{minB}, \text{maxB}])$ respectively. Second, the time series data are generated according to the so defined AR(1) model.

Usage

```
out<-simulAR1(p,n,edgeProp,minA,maxA,minB,maxB,minSig,maxSig)
```

Arguments

<code>p</code>	the desired dimension of the multivariate time series.
<code>n</code>	the desired length of the time serie.
<code>edgeProp</code>	the desired proportion of non zero coefficient in the AR transition matrix.
<code>minA</code>	the minimum value for matrix A elements generation.
<code>maxA</code>	the maximum value for matrix A elements generation.
<code>minB</code>	the minimum value for matrix B elements generation.
<code>maxB</code>	the maximum value for matrix B elements generation.
<code>minSig</code>	the minimum value for the diagonal of covariance matrix S generation.
<code>maxSig</code>	the maximum value for the diagonal of covariance matrix S generation.

Value

A list with `out$data` a matrix, with `n` rows (`=length`) and `p` columns (`=dimension`), containing the generated time series, `out$A` the AR generated matrix A (`p` x `p`), `out$B` the AR generated vector B (`p`), `out$sig` the generated diagonal (`p`) of covariance matrix S .

Author(s)

Lebre Sophie (<http://stat.genopole.cnrs.fr/~slebre>).

Examples

```
#generate AR(1) time series
AR<-simulAR1(p=10,n=50,edgeProp=0.02,minA=0.5,maxA=1.5,minB=0,maxB=1,minSig=0.1,
maxSig=0.8)
```

arth800line

Arabidopsis Thaliana temporal gene expression data

Description

This data set describes the temporal expression of 800 genes of *A. thaliana* during the diurnal cycle. The data are in line, that is 2 repeated measurements time series are displayed one after the other, separated by a NA value. The 800 genes are a subset of the data presented in Smith et al. (2004) selected for periodicity according to the method implemented in the R package GeneCycle (<http://strimmerlab.org/software/genecycle/>).

Usage

```
data(arth800line)
```

Format

matrix with 800 columns (=genes) and 23 rows (rows 1 to 11 for the first measurement time series, row 12 is NA and rows 13 to 23 for the second experiment time series).

Author(s)

Lebre Sophie (<http://stat.genopole.cnrs.fr/~slebre>).

Source

The microarray experiments were performed in the laboratory of S. Smith (Edinburgh). The data are available from the NASCArrays database at <http://affymetrix.arabidopsis.info/> under experiment reference number NASCARRAYS-60.

References

Smith et al. 2004. Diurnal changes in the transcriptom encoding enzymes of starch metabolism provide evidence for both transcriptional and posttranscriptional regulation of starch metabolism in Arabidopsis leaves. *Plant Physiol.* 136: 2687-2699.

Examples

```
# load G1DBN library
library(G1DBN)

# load data set
data(arth800line)

# plot first ten time series
plot(1:23,arth800line[,1],type="l",ylim=c(5,12))
for (i in 2:10)lines(1:23,arth800line[,i],col=i)
```


Bibliography

- [AB94] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:641–647, 1994.
- [AD99] C. Andrieu and A. Doucet. Joint bayesian model selection and estimation of noisy sinusoids via reversible jump mcmc. *IEEE Trans. on Signal Processing*, 47(10):2667–2676, 1999.
- [ADD01] C. Andrieu, P. Djurić, and A. Doucet. Model selection by mcmc computation. *Signal Processing*, pages 19–37, 2001.
- [BdlFM02] P. Brazhnik, A. de la Fuente, and P. Mendes. Gene networks: how to put the function in genomics. *Trends in Biotechnology*, 20:467–472, 2002.
- [Ber95] A. Berchtold. Autoregressive modeling of Markov chains. In *Statistical Modelling: Proceedings of the 10th International Workshop on Statistical Modelling*, pages 19–26. Springer-Verlag, 1995.
- [Ber01] A. Berchtold. Estimation in the Mixture Transition Distribution model. *Journal of Time Series Analysis*, 22(4):379–397, 2001.
- [BFG⁺05] M.J. Beal, F.L. Falciani, Z. Ghahramani, C. Rangel, and D. Wild. A bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21:349–356, 2005.
- [BJ05] Ziv Bar-Joseph. Analyzing time series gene expression data. *Bioinformatics*, 20(16):2493–503, 2005.
- [BM93] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing. Optical Engineering*, 34:641–647, 1993.
- [BR02] André Berchtold and Adrian E. Raftery. The Mixture Transition Distribution model for high-order Markov chains and non-gaussian time series. *Statistical Science*, 17:328–356, 2002.
- [BR04] Pierre-Yves Bourguignon and David Robelin. In proceedings of the journées ouvertes biologie informatique mathématique, montréal, 2004.
- [BTS⁺00] A. J. Butte, P. Tamayo, D. Slonim, T. R. Golub, and I. S. Kohane. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proc Natl Acad Sci U S A*, 97(22):12182–12186, October 2000.
- [BW99] P. Bühlmann and A.J. Wyner. Variable length Markov chains. *Annals of Statistics*, 27, 1999.

- [Chu89] G. Churchill. Stochastic models for heterogeneous DNA sequences. *Bulletin of Mathematical Biology*, 51(1):79–94, 1989.
- [CR06] R. Castelo and A. Roverato. Graphical model search procedure in the large p and small n paradigm with applications to microarray data. *Journal of Machine Learning Research*, 7:2621–2650, 2006.
- [CW96] D R. Cox and N. Wermuth. *Multivariate dependencies: Models, analysis and interpretation*. Chapman and Hall, London, 1996.
- [DEKM99] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, July 1999.
- [DJ02] H. De Jong. Modeling and simulation of genetic regulatory systems: a literature review. *Computational Biology*, 9:437–467, 2002.
- [DIFBHM04] A. De la Fuente, N. Bing, I. Hoeschele, and P. Mendes. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20:3565–3574, 2004.
- [DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B.*, 39:1–38, 1977.
- [DRD05] Paul Delmar, Stéphane Robin, and Jean-Jacques Daudin. Varmixt: efficient variance modelling for the differential analysis of replicated gene expression data. *Bioinformatics*, 21(4):502–508, 2005.
- [Edw95] D. Edwards. *Introduction to Graphical Modelling*. Springer-Verlag, New York, 1995.
- [Efr05] B. Efron. Local false discovery rates. *Technical Report number. Dept. of Statistics, Stanford University.*, 2005.
- [EHJT04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.
- [FG87] G. Fichant and C. Gautier. Statistical method for predicting protein coding regions in nucleic acid sequences. *Computer applications in the biosciences : CABIOS.*, 3:287–295, 1987.
- [FLNP00] N. Friedman, M. Linial, I. Nachman, and D. Pe’er. Using bayesian networks to analyse expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.
- [FMR98] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In *Proceedings of the 14th conference on the Uncertainty in Artificial Intelligence*, pages 139–147, SM, CA, USA, Morgan Kaufmann, 1998.
- [Fox02] J. Fox. *An R and S-Plus companion to applied regression*. Sage Publications, Thousand Oaks, CA, USA, 2002.
- [GF05] T. S. Gardner and J. J. Faith. Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2:65–88, 2005.
- [Gre95] P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

- [Gre05] A. Grelot. Estimation bayésienne d'un modèle MTD - MSc Report available at <http://stat.genopole.cnrs.fr/sg/members/slebre/rapportadeline.pdf/view>, 2005.
- [IGM02] S. Imoto, T. Goto, and S. Miyano. Estimation of genetic networks and functional structures between genes by using bayesian networks and nonparametric regression. In *Pacific Symposium on Biocomputing 7*, pages 175–186, 2002.
- [IKG⁺03] S. Imoto, S. Kim, T. Goto, S. Aburatani, K. Tashiro, S. Kuhara, and S. Miyano. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *Journal of Bioinformatics Computational Biology*, 2:231–252, 2003.
- [KIM03] S. Kim, S. Imoto, and S. Miyano. Inferring gene networks from time series microarray data using dynamic bayesian networks. *Briefings in Bioinformatics*, 4(3):228, 2003.
- [KIM04] S. Kim, S. Imoto, and 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.
- [Lau96] S. L. Lauritzen. *Graphical models*. Oxford Statistical Science Series, 1996.
- [Lau98] S. L. Lauritzen. *Graphical models. Repr.* Oxford Statistical Science Series. 17., 1998.
- [LDLK⁺05] A. Lucau-Danila, G. Lelandais, Z. Kozovska, V. Tanty, T. Delaveau, F. Devaux, , and C. Jacq. Early expression of yeast genes affected by chemical stress. *Mol Cell Biol.*, 25(5):1860–1868, 2005.
- [Leb07] S. Lebre. Inferring dynamic genetic networks with low order independencies. *Submitted to SAGMB*, 2007.
- [LK90] W.K. Li and Michael C.O. Kwok. Some results on the estimation of a higher order Markov chain. *Commun. Stat. Simulat.*, 19(1):363–380, 1990.
- [LRR⁺02] 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, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in *saccharomyces cerevisiae*. *Science*, 298(5594):799–804, 2002.
- [Mee95] C. Meek. Strong completeness and faithfulness in bayesian networks. In *Proc. of the 11th Annual Conference on Uncertainty in Artificial Intelligence*, SF, CA, USA, Morgan Kaufmann Publishers, 1995.
- [MK04] P. M. Magwene and J. Kim. Estimating genomic coexpression networks using first-order conditional independence. *Genome Biology*, 5(12), 2004.
- [MM99] K. Murphy and S. Mian. Modelling gene expression data using dynamic bayesian networks. *Technical report, Computer Science Division, University of California, Berkeley, CA.*, 1999.
- [Mur97] F. Muri. *Comparaison d'algorithmes d'identification de chaînes de Markov cachées et application à la détection de régions homogènes dans les séquences d'ADN*. PhD thesis, Université d'Evry-Val-d'Essonne, 1997.
- [Mur01] K. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 33, 2001.

- [Nic03] P. Nicolas. *Mise au point et utilisation de modèles de chaînes de Markov cachées pour l'étude des séquences d'ADN*. PhD thesis, Université d'Evry-Val-d'Essonne, 2003.
- [OGP02] I. M. Ong, J. D. Glasner, and D. Page. Modelling regulatory pathways in e. coli from time series expression profiles. *Bioinformatics*, 18(Suppl 1):S241–S248, 2002.
- [ORS07] R. Opgen-Rhein and 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.
- [Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. SF, CA, USA, Morgan Kaufmann Publishers, 1988.
- [PRM⁺03] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. d'Alché Buc. Gene networks inference using dynamic bayesian networks. *Bioinformatics*, 19(Suppl 2):S138–S148, 2003.
- [Raf85] A. E. Raftery. A model for high-order Markov chains. *Journal of the Royal Statistical Society. Series B*, 47(3):528–539, 1985.
- [RAG⁺04] C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani. Modeling t-cell activation using gene expression profiling and state-space models. *Bioinformatics*, 20(9):1361–1372, 2004.
- [RHISE07] A. Rao, A. O. Hero III, D. J. States, and J. D. Engel. Inferring time-varying network topologies from gene expression data. *EURASIP Journal on Bioinformatics and System Biology - Special Issue on Gene Networks*, 2007.
- [RRS02] M. Ronen, R. Rosenberg, and Shraiman. Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics. *PNAS*, 99:10555:10560, 2002.
- [RT94] Adrian E. Raftery and Simon Tavaré. Estimation and modelling repeated patterns in high order Markov chains with the Mixture Transition Distribution model. *Journal of the Royal Statistical Society Applied Statistics*, 43(1):179–199, 1994.
- [SBH⁺01] I. Simon, J. Barnett, N. Hannett, C. Harbison, N. Rinaldi, J. Volkert, T. and Wyrick, J. Zeitlinger, Gifford D. K., Jaakkola T. S., and R. A. Young. Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, 106:697–708, 2001.
- [Sch78] Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.
- [SFC⁺04] S. M. Smith, D. C. Fulton, T. Chia, D. Thorneycroft, A. Chapple, H. Dunstan, C. Hylton, S. C. Zeeman, and A. M. Smith. Diurnal changes in the transcriptome encoding enzymes of starch metabolism provide evidence for both transcriptional and posttranscriptional regulation of starch metabolism in arabidopsis leaves. *Plant Physiol.*, 136(1):2687–2699, 2004.
- [SGS93] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction and search*. Springer Verlag, New York (NY), 1993.
- [SI04] N. Sugimoto and H. Iba. Inference of gene regulatory networks by means of dynamic differential bayesian networks and nonparametric regression. *Genome Informatics*, 15(2):121–130, 2004.

- [SK99] R. Somogyi and H. Kitano. Gene expression and genetic networks. *Pacific Symposium in Biocomputing*, pages 3–4, 1999.
- [SKFW03] R. Steuer, J. Kurths, O. Fiehn, and W. Weckwerth. Observing and interpreting correlations in metabolomic networks. *Bioinformatics*, 19(8):1019–1026, 2003.
- [SS05a] J. Schäfer and K. Strimmer. An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21:754–764, 2005.
- [SS05b] J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(32), 2005.
- [SSDB95] M. Schena, D. Shalon, R. Davis, and P. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 70:467–470, 1995.
- [SSZ⁺98] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Mol Biol Cell*, 9(12):3273–3297, 1998.
- [SWDS03] M.A. Suchard, R.E. Weiss, K.S. Dorman, and J.S. Sinsheimer. Inferring spatial phylogenetic variation along nucleotide sequences: a multiple change-point model. *Journal of the American Statistical Association*, 98:427–437, 2003.
- [SYS03] G. K. Smyth, Y. H. Yang, and T. Speed. Statistical issues in cDNA microarray data analysis. *Methods in Molecular Biology*, 224:111–136, 2003.
- [TH02a] H. Toh and K. Horimoto. Inference of a genetic network by a combined approach of cluster analysis and graphical gaussian modeling. *Bioinformatics*, 18:287–297, 2002.
- [TH02b] H. Toh and K. Horimoto. System for automatically inferring a genetic network from expression profiles. *J. Biol. Physics*, 28:449–464, 2002.
- [Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58:267–288, 1996.
- [TM06] Miguel C. Teixeira and Pedro Monteiro. The YEASTRACT database: a tool for the analysis of transcription regulatory associations in *saccharomyces cerevisiae* [<http://www.yeasttract.com>]. *Nucleic Acids Research*, 34:D446–D451, 2006.
- [WB06] A. Wille and P. Bühlmann. Low-order conditional independence graphs for inferring genetic networks. *Statist. Appl. Genet. Mol. Biol*, 4(32), 2006.
- [WFS04] S. Wichert, K. Fokianos, and K. Strimmer. Identifying periodically expressed transcripts in microarray time series data. *Bioinformatics*, 20:5–20, 2004.
- [Whi90] J. Whittaker. *Graphical models in applied multivariate statistics*. Wiley, NY, 1990.
- [WK00a] P. J. Waddell and H. Kishino. Cluster inference methods and graphical models evaluated on nci60 microarray gene expression data. *Genome Informatics*, 11:129–140, 2000.
- [WK00b] P. J. Waddell and H. Kishino. Correspondence analysis of genes and tissue types and finding genetics links from microarray data. *Genome Informatics*, 11:83–95, 2000.

- [WLL04] G. H. Wei, D. P. Liu, and C. C. Liang. Charting gene regulatory networks: strategies, challenges and perspectives. *Biochemical Journal*, 381:1–12, 2004.
- [WMH03] J. Wang, O. Myklebost, and E. Hovig. Mgraph: graphical models for microarray data analysis. *Bioinformatics*, 19(17):2210–2211, 2003.
- [Wu83] C. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.
- [WYS03] X. Wu, Y. Ye, and K. R. Subramanian. Interactive analysis of gene interactions using graphical gaussian model. *ACM SIGKDD Workshop on Data Mining in Bioinformatics*, 3:63–69, 2003.
- [WZK04] F. X. Wu, W. J. Zhang, and A. J. Kusalik. Modeling gene expression from microarray expression data with state-space equations. In *Pacific Symposium on Biocomputing*, pages 581–592, 2004.
- [WZV⁺04] A. Wille, P. Zimmermann, E. Vranova, A. Fürholz, O. Laule, and S. Bleuler. Sparse graphical gaussian modeling for genetic regulatory network inference. *Genome Biol*, 5(11), 2004.
- [YBDS02] Y. H. Yang, M. J. Buckley, S. Dudoit, and T. P. Speed. Comparison of methods for image analysis on cDNA microarray data. *Journal of Computational and Graphical Statistics*, 11:108–136, 2002.
- [YDLS01] Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed. Normalization for cDNA microarray data. In *Microarrays: Optical Technologies and Informatics*, volume 4266 of Proceedings of SPIE, 2001.
- [ZC05] M. Zou and S. D. Conzen. A new dynamic bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 21(1):71–79, 2005.