MASTER EN SCIENCES MATHÉMATIQUES À FINALITÉ DIDACTIQUE

Étude d'une nouvelle méthode d'identification de réseaux basée sur l'opérateur de Koopman
applications aux réseaux de régulation génétique.

Devillers, Juliette

Award date:
2021

Awarding institution:
Universite de Namur

UNIVERSITÉ DE NAMUR

Faculté des Sciences

ÉTUDE D'UNE NOUVELLE MÉTHODE D'IDENTIFICATION DE RÉSEAUX BASÉE SUR L'OPÉRATEUR DE KOOPMAN: APPLICATION AUX RÉSEAUX DE RÉGULATION GÉNÉTIQUE

Mémoire présenté pour l'obtention

du grade académique de master en sciences mathématiques

Juliette Devillers

Juin 2021

Remerciements

Je tiens à remercier particulièrement mon promoteur Alexandre Mauroy pour son suivi régulier et ses précieux conseils. Je le remercie également pour toutes les suggestions et corrections qu'il m'a apportées lors de la rédaction.

Ce mémoire est l'aboutissement de 5 années de travail. C'est pourquoi j'adresse également mes remerciements aux autres enseignants du département de mathématiques qui m'ont permis d'évoluer au travers de ces années.

Je remercie aussi mes proches pour m'avoir soutenue et encouragée tout au long de ma formation et plus particulièrement lors de la réalisation de ce mémoire. Ils ont été un soutient bien nécessaire.

RÉSUMÉ

Dans ce mémoire, nous étudierons une nouvelle méthode d'identification de réseau qui repose sur l'opérateur de Koopman. Celle-ci présente deux avantages majeurs : elle est valide pour les systèmes non linéaires et lorsque le nombre de données temporelles est faible. De plus, un résultat de convergence permet d'attester de son efficacité dans le cas théorique.

Nous identifierons plus spécifiquement des réseaux de régulation génétique. Notre méthode d'identification a été testée sur les données du concours *DREAM-4 in silico network challenge* qui visent à simuler des données réelles issues de la génétique. Les résultats obtenus sont supérieurs à ce que nous attendrions d'une identification aléatoire, mais ceux-ci reste assez faibles. La difficulté de la tâche s'explique par la complexité des systèmes de régulation génétique (grande dimension, systèmes non linéaires, etc). Finalement, nous proposerons un modèle pour le réseau de régulation de l'horloge circadienne de la plante Arabidopsis thaliana.

Mots clés : identification de réseaux, système différentiel, opérateur de Koopman, réseaux de régulation génétiques

ABSTRACT

In this master thesis, we will be studying a new method for network identification, based on the Koopman operator. This method features two major advantages : it is valid for non-linear systems and doesn't require a large numer of data points. Furthermore, we can show its efficacy in the theoretical case.

More specifically, we will be studying gene regulatory network. Our identification method was test using data from the *DREAM-4 in silico network challenge*, a dataset aiming to simulate real-world data from genetics. Obtained results were surpassing what can be expected out of a random identification, but stayed relatively modest. The difficulty of the task can be explained by the complexity of such genetic regulation systems (high dimensionality, non-linear systems, etc). In the end we propose a model for gene regulatory network of the circadian clock of Arabidopsis thaliana.

Keywords : networks identification, differential system, Koopman operator, gene regulatory network

TABLE DES MATIÈRES

INTRODUCTION

Pour pouvoir étudier les organismes vivants, les biologistes travaillent à plusieurs échelles : moléculaire, cellulaire, niveau de l'organisme ou l'ensemble d'une population. À chacun de ses niveaux, de nombreux éléments entrent en jeu. Pour comprendre le fonctionnement de l'ensemble, il est donc nécessaire de déterminer comment interagissent les différents composantes, c'est-à-dire définir un réseau. Ces représentations servent, par exemple, à expliquer comment des neurones interagissent entre eux ou les interactions entre différentes espèces d'un même éco-système.

Construire le réseau d'un système biologique permet, dans un premier temps, d'expliquer son fonctionnement et de l'analyser. Dans un second temps et lorsque le modèle est suffisamment précis, il permet de réaliser des prédictions et de déterminer quelle serait la réaction du système face à une nouvelle substance ou à l'introduction d'une espèce. À terme, l'objectif est de parvenir à contrôler les systèmes étudiés.

Nous travaillerons plus spécifiquement sur les réseaux de régulation génétique. Ils représentent les interactions entre les différents gène d'un même individu. L'étude de ceux-ci s'est particulièrement développée grâce aux puces à ADN. Elles permettent de mesurer le taux d'activation des gènes. Ces données sont utilisées pour construire les réseaux de régulation génétique. Malgré l'apport de cette nouvelle technologie, identifier ces réseaux reste une tâche d'une grande complexité notamment car les éléments intervenants dans le système sont très nombreux.

Dû au grand nombre d'éléments constitutifs de ces réseaux, leur modélisation est en général sous-déterminée : il y a plus d'éléments à identifier N que de données à disposition K. Dans ce mémoire, nous proposerons une méthode d'identification de réseau pouvant être utilisée lorsque $K \leq N$. Celle-ci repose sur l'opérateur de Koopman et sur des méthodes linéaires. Elle est pourtant valide pour identifier des systèmes non linéaires.

Dans le premier chapitre, nous expliquerons plus précisément la notion de réseaux de régulation génétique et les difficultés liées à son identification. Nous évoquerons également des modélisations possibles et des méthodes d'identification. En pratique, pour

identifier les réseaux, nous utiliserons une nouvelle méthode basée sur l'opérateur de Koopman et proposée dans les articles [21] et [22]. Celle-ci sera présentée en détails au chapitre 2. L'identification nécessite également de réaliser une régression et deux méthodes seront également proposées. L'une est basée sur les arbres décisionnels et l'autre est une régression linéaire avec une contrainte sur les coefficients.

Ensuite, cette méthode sera utilisée pour identifier des réseaux. Dans un premier temps, au chapite 3, nous travaillerons sur les données synthétiques du concours *DREAM-4 In silico network challenge*. Ces données représentent le taux d'ARNm contenus dans les gènes. Elles ont été construites de la manière la plus réaliste possible. Elle nous permettront d'évaluer la méthode et de pouvoir comparer les résultats avec les réseaux réels. Nous évoquerons également des pistes d'amélioration pour la méthode d'identification proposée. Finalement, dans le chapitre 4, nous identifierons le réseau de régulation génétique de l'horloge circadienne de la plante Arabidopsis thaliana. Le génome de cette fleur est particulièrement étudié notamment parce-qu'elle est une des premières espèces à avoir eu son génome complètement séquencé. Cela permettra de pouvoir comparer le modèle obtenu avec d'autres résultats.

CHAPITRE 1
RÉSEAUX ET IDENTIFICATION

Dans ce chapitre, nous introduirons l'identification de réseau et plus spécifiquement dans le contexte des réseaux de régulation génétique. Dans un premier temps, nous développerons le cadre de travail et la notion de réseau de régulation génétique, nous évoquerons aussi les difficultés de ce type de problème. Ensuite, nous discuterons de deux types de représentation de réseaux très utilisées : les graphes et les équations différentielles. Nous présenterons aussi des méthodes d'identification. Finalement, nous évoquerons des critères d'évaluation pour ces modèles.

1.1 Réseaux en biologie

1.1.1 Contexte

Les systèmes complexes sont fondamentaux en biologie, en effet, la dynamique d'un système est influencée par un ensemble d'interactions dépendantes entre elles et non par des mécanismes isolés [1]. L'étude de ces réseaux a un double intérêt, elle permet, en premier lieu, de pouvoir décrire les mécanismes (leurs acteurs, leurs interactions,...) et donc élargir la connaissance du vivant. La compréhension du fonctionnement des systèmes biologiques a de nombreuses applications dans le domaine médical puisqu'elle donne accès à une compréhension plus fine des mécanismes d'un traitement ou d'une pathologie. Par exemple, l'identification de gènes spécifiques intervenant dans une maladie permet un diagnostic et une étude des prédispositions plus aisées [1].

Dans un second temps, lorsque la modélisation est suffisamment précise, elle permet également de réaliser des prédictions. Celles-ci ont également de nombreuses applications. Elles peuvent permettre de remplacer, par des estimations, des expériences coûteuses ou avec des effets indésirables [14]. Par exemple, des chercheurs étudient, par le biais de modèles, la réaction théorique de systèmes face à des facteurs extérieurs (virus, bactéries, drogues, traitement, etc) [1]. L'objectif à long terme est de suffisamment connaître ces systèmes pour pouvoir les maîtriser.

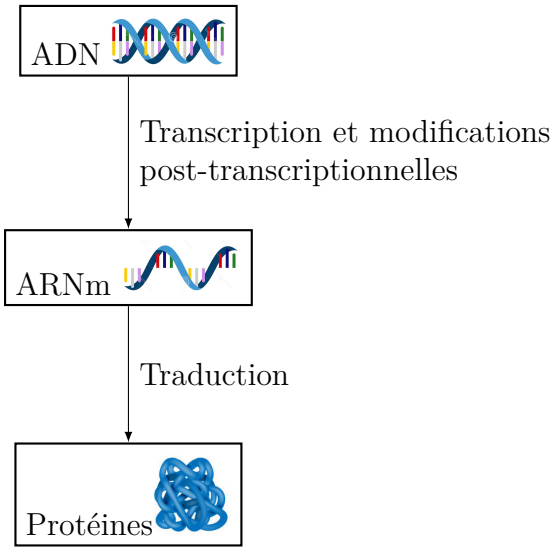


FIGURE 1.1 – Synthèse des protéines

1.1.2 Réseaux de régulation génétique

L'avancée de la recherche de ces dernières décennies a permis de séquencer complètement les gènes de divers organismes. Aujourd'hui les scientifiques vont plus loin et recherchent les interactions qui ont lieu entre ces différents gènes. Pour définir le fonctionnement des réseaux de régulation génétique, il faut d'abord évoquer le processus de synthétisation des protéines (aussi appelé expression génétique) illustré à la figure 1.1. La description de ce processus biologique et la définition des réseaux de régulation génétique sont basés sur les références [1], [4] et [12].

Les gènes sont des segments d'ADN qui permettent de coder une molécule en particulier, des protéines principalement. Celles-ci réalisent ensuite diverses fonctions au sein de l'organisme. La première étape pour synthétiser des protéines est la transcription qui a pour but de "copier" la séquence de l'ADN grâce à l'ARN. La molécule d'ARN est le complémentaire d'un des deux brins de l'ADN et, contrairement à ce dernier, l'ARN est composé d'un seul brin. Ensuite ce brin d'ARN subit diverses modifications appelées post-transcriptionnelles, notamment la suppression des segments non nécessaires et il devient de l'ARN messager (ARNm). Ce dernier permet de transporter l'information hors du noyau de la cellule. Finalement les molécules d'ARNm sont traduites, c'est-à-dire que des molécules de protéines sont synthétisées à partir du code contenu dans l'ARNm.

Les réseaux de régulation génétique (RRG) représentent la manière dont chaque gène régule (positivement ou négativement) l'expression des autres gènes. Par "expression de gène", nous désignons le fait que le code contenu dans le gène est utilisé pour synthétiser une protéine. En pratique, les gènes n'interagissent pas directement entre eux, mais au travers de protéines comme illustré à la figure 1.2 : la protéine produite par le gène A influence l'expression du gène B.

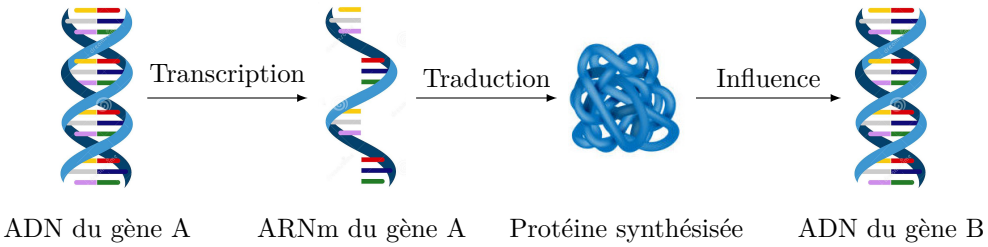


FIGURE 1.2 – Processus d'influence entre deux gènes

Les RRG nécessitent l'étude de plusieurs niveaux (ADN, ARNm et protéines) et les différentes composantes doivent être analysées comme un tout et non parallèlement, ce qui rend leur étude complexe. Généralement ce sont plutôt les réseaux de régulation transcriptionnelle (RRT) qui sont utilisés. Ceux-ci sont une simplification des réseaux de régulation génétique. Ils tiennent compte uniquement de la régulation de l'ARNm et supposent que l'ARNm d'un gène influence directement celui d'un autre. Les RRT sont intéressants d'un point de vue pratique puisque le taux d'ARNm est facilement obtenu grâce aux puces à ADN, l'inférence des RRT est donc favorisée. Les deux termes (RRG et RRT) sont régulièrement confondus dans la littérature et pour la suite, nous conserverons le terme de réseau de régulation génétique plus couramment utilisé, même si nous utiliserons l'hypothèse simplificatrice des RRT.

Aujourd'hui, les puces à ADN permettent d'estimer le niveau d'ARN messager dans des dizaines de milliers de gènes à la fois [18]. Cette technologie est la plus couramment utilisée dans le cadre de l'étude de l'expression de gènes [1]. Les mesures réalisées via les puces à ADN sont peu coûteuses et permettent d'obtenir des résultats facilement [4]. L'identification des RRG est basée sur le taux d'ARNm mesuré dans chaque gène.

1.1.3 Difficultés

Dans les sections précédentes, nous avons évoqué la complexité des réseaux biologiques et plus particulièrement des RRG. Il est donc logique que la construction de ces réseaux ne soit pas une tâche aisée et nous allons discuter des difficultés pour les identifier. Celles-ci sont, d'une part, liées à la nature même des systèmes étudiés et d'autre part, au type de données utilisées.

Tout d'abord, contrairement à des systèmes informatiques, nous ne pouvons connaître le fonctionnement des réseaux biologiques que de manière expérimentale. En effet, ceux-ci fonctionnent comme des boites noires et il faut déterminer comment le processus évolue uniquement à partir d'observations [18]. Par ailleurs les systèmes biologiques sont complexes parce qu'ils contiennent en général de très nombreux éléments. À titre d'exemple, le génome humain est composé d'environ 25000 gènes [1]. Les études réalisées portent donc en général sur un nombre restreint de gènes ayant une spécificité commune.

Les données mises à disposition sont par ailleurs limitées. En effet, bien que les puces à ADN soient facile d'usage, les mesures sous souvent destructrices et nécessitent de

nombreuses manipulations. Pour obtenir des données temporelles il faut lancer simultanément différentes cultures et utiliser chacune d'elle pour obtenir une seule donnée temporelle [1]. Les données sont alors dites pseudo-temporelles. Étant donné le grand nombre de gènes, même en se concentrant sur un nombre restreint, le problème d'identification des RRG est en général sous-déterminé.

Par rapport aux données, il y a également le problème du bruit. Il est d'une part lié aux appareils qui permettent de réaliser les mesures comme les puces à ADN et d'autre part aux conditions expérimentales. Les mesures proviennent généralement de moyennes effectuées sur plusieurs cellules et plusieurs cultures sont utilisées, ce qui implique de la variabilité dans les données [1].

Un autre obstacle, c'est la nature même des systèmes biologiques : ils sont stochastiques et fortement non linéaires [18]. Les modèles stochastiques et/ou non linéaires sont plus difficiles à estimer et à manipuler.

1.2 Modélisation mathématique des réseaux

1.2.1 Généralités

Déterminer la structure d'un réseau est essentiel dans divers domaines comme en économie, en ingénierie ou en biologie comme cela vient d'être présenté. L'objectif est le suivant : pour pouvoir étudier les réseaux de régulation génétique, nous cherchons à construire un modèle mathématique exprimant la dynamique d'un système. L'identification d'un réseau s'effectue en deux temps.

1. **Identification des "éléments"**
   Avant de rechercher les interactions entre les différents éléments, il faut déterminer les nœuds du réseau. Pour cela, il faut choisir un niveau de représentation : il est possible de considérer chaque élément de manière distincte ou alors d'en regrouper certains selon des critères bien définis. Dans le cadre de la construction de réseaux génétiques que nous réaliserons, chaque gène pris en compte sera considéré comme un élément du réseau.

2. **Identification des interactions entre les éléments**
   L'expression des interactions peut prendre des formes très variées, des plus simples au plus complexes. C'est pourquoi, il existe de nombreux modèles pour désigner un réseau. Ils se différencient notamment par la nature des liens entre les éléments mais aussi par la quantité d'informations données : certains modèles sont plus précis que d'autres. Néanmoins, il faut garder à l'esprit que, en général, au plus un modèle délivre des informations, au plus il nécessite de données pour le construire. Les modèles suffisamment précis ont l'avantage de pouvoir réaliser des prédictions.

Dans cette section, deux modèles en particulier seront présentés : les graphes et les équations différentielles. Dans la section suivante, nous verrons des méthodes d'identification pour ces réseaux.

1.2.2 Modélisation statique

La modélisation via des graphes est la plus intuitive et visuelle. Dans sa version la plus simple, le graphe donne une description qualitative d'un système à un instant donné. Cela signifie que c'est une description statique du système. Chaque élément du système est représenté par un nœud et le lien entre deux éléments est représenté par une arrête. Cette première version apporte seulement une description qualitative du modèle, mais permet tout de même de répondre à des questions simples [28].

Lorsque le nombre de nœuds du réseau est important, il est plus simple de représenter le graphe via la matrice d'adjacence. Cette matrice est de dimension $n \times n$ où n est le nombre d'éléments dans le réseau et elle est définie comme suit

$$a_{ij} = \begin{cases} 1 \text{ si les éléments } i \text{ et } j \text{ ont un lien} \\ 0 \text{ sinon.} \end{cases}$$

Cette matrice est symétrique et ne permet donc pas d'identifier le sens de l'interaction. Dans le contexte de la biologie, nous cherchons le sens de ces liens, c'est pourquoi l'usage d'arrêtes directionnelles est favorisé. La matrice d'adjacence A est alors définie par

$$a_{ij} = \begin{cases} 1 \text{ si l'élément } i \text{ influence l'élément } j \\ 0 \text{ sinon.} \end{cases} \tag{1.1}$$

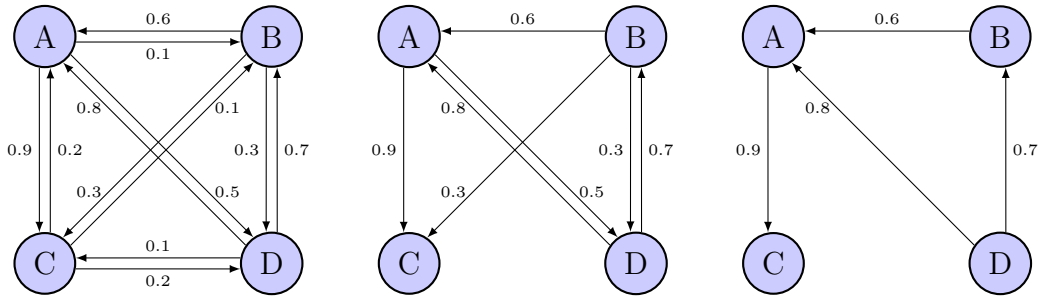
Il est à nouveau possible de complexifier ce modèle en utilisant des arrêtes pondéré. Ainsi le modèle devient quantitatif et permet de mesurer le taux d'influence entre les différents éléments. Nommons la matrice d'adjacence pondérée par $\tilde{A}$, elle est définie par

$$\tilde{a}_{ij} = \omega_{ij} \tag{1.2}$$

où ω_{ij} est une mesure du lien de i vers j.

En général, les méthodes d'identification permettent d'obtenir des modèles quantitatifs et attribuent donc un poids pour chaque arrête. Dans ce cas, les graphes obtenus sont complètement connectés avec des arrêtes dont le poids varie. Pour pouvoir obtenir un graphe moins connecté et détecter les nœuds les plus importants, il suffit alors de choisir un seuil à partir duquel on considérera qu'il y a bien un lien entre les deux nœuds. Le choix de ce seuil est donc essentiel : s'il est trop bas, le réseau obtenu risque de contenir des arrêtes en trop et s'il est trop important, le modèle pourrait contenir trop peu de liens. Il faut donc à la fois minimiser le nombre de faux négatifs et de faux positifs.

L'influence du choix du seuil est illustré à la figure 1.3. Sans seuil fixé, le graphes est complètement connecté. Ensuite, au plus le seuil est grand, au plus le nombre d'arrêtes est faible et permet de mettre en avant les interactions les plus importantes. Pour évaluer un modèle, il faudra donc tenir compte du choix de ce seuil. Dans la section 1.4, nous verrons deux méthodes d'évaluation justement basées sur l'influence de ce seuil.



(a) Pas de seuil fixé (b) Seuil=0.3 (c) Seuil=0.6

FIGURE 1.3 – Influence du seuil

1.2.3 Modélisation dynamique

La modélisation sous forme d'un graphe représenté par sa matrice d'adjacence est statique : elle ne dépend pas du temps et représente de manière figée les interactions. Il est également possible de modéliser les réseaux sous une forme dynamique pour exprimer la dépendance au temps. La modélisation dynamique la plus utilisée dans le contexte biologique est la modélisation via les équations différentielles [28]. Elles permettent une description quantitative et instantanée du réseau. Elles délivrent donc des informations précises et donne également lieu à des prédictions.

Nommons x_i la variable associée au nœud i, dans notre cas, x_i représente le taux d'ARNm dans le gène i. Le modèle différentiel est alors de la forme

$$\dot{x_i} = f_i(x_1, ..., x_n, t, \Theta), \qquad \forall i = 1, ..., n \tag{1.3}$$

avec f_i la fonction exprimant la dérivée de x_i en fonction des autres variables, t le temps et $\Theta = (\theta_1, ..., \theta_N)$ l'ensemble des paramètres.

Dans la pratique, nous modéliserons les réseaux étudiés sous forme d'équations différentielles. Il est ensuite possible de simplifier ce modèle en le ramenant sous forme de graphe. Cela peut s'effectuer au travers d'une régression comme expliqué à la section 2.2. Cette représentation permet d'avoir un modèle plus simple à évaluer.

1.3 Méthodes d'identification

Dans cette section, nous évoquerons différentes stratégies pour construire des réseaux sous forme de graphe ou d'équations différentielles à partir de données temporelles.

1.3.1 Méthodes pour les graphes

Une première approche assez naïve pour former un graphe est de construire des arrêtes pondérées par le coefficient de corrélation linéaire [12]. Cette méthode s'implémente facilement et est rapide mais elle présente plusieurs désavantages. Tout d'abord, elle recherche uniquement les corrélations deux à deux. Aussi, le coefficient de corrélation est linéaire et n'est donc pas efficace dans le cadre de réseaux non linéaires.

Finalement, ce coefficient est symétrique et ne permet pas de diriger les arrêtes.

Il est également possible de construire des modèles en utilisant la régression linéaire. Nous cherchons donc à exprimer l'élément x_j par

$$x_j = \sum_{i=1,\neq i}^{n} \alpha_{ij} x_i \qquad \forall i = 1, ..., n$$

où α_{ij} est le facteur qui représente l'influence du nœud i sur le nœud j et n le nombre de nœud . Les coefficients α_{ij} permettent de construire la matrice d'adjacence. Cette méthode mène à des graphes pondérés et directionnels et qui en plus permettent de considérer la corrélation globale entre les éléments et non uniquement deux à deux. Il reste par contre lui aussi limité puisqu'il n'est valide que dans le cas des systèmes linéaires, or ce n'est pas le cas des systèmes de régulation génétique. De plus, il ne tient pas compte de l'aspect dynamique puisque la variable temporelle t n'influence pas le modèle.

Il est possible de réaliser des régressions plus complexes (non linéaires et avec plus de fonctions de base). Il faut alors supposer les fonctions avec lesquelles la régression va être effectuée. Dans le cas où les fonctions de base sont nombreuses, la régression avec pénalité, comme la méthode lasso évoquée plus tard, permet d'imposer que certains coefficients soient nuls.

Si l'on veut construire un graphe qui ne présuppose pas la forme des interactions, les méthodes de régression utilisant les arbres décisionnels constituent une bonne alternatives. Celles-ci réalisent une régression via une fonction constante par morceau. Un tel algorithme sera présenté dans le chapitre 2 et utilisé lors des simulations.

1.3.2 Méthodes pour les systèmes d'équations différentielles

Pour obtenir un modèle différentiel de la forme (1.3), il faut fixer les fonctions f_i qui représentent la dynamique du système. Généralement, les méthodes supposent que les f_i sont de lois connues et estiment les paramètres Θ. En pratique, de tels connaissances a priori sur la dynamique du système ne sont pas toujours possibles. L'estimation des paramètres Θ est en général réalisée grâce à un problème d'optimisation [5]. Dans ce cas, il faut définir une fonction score $J(\Theta)$ qui devra être minimisée.

L'approximation des paramètres nécessite en général un nombre de données temporelles (K) plus important que le nombre de paramètres à estimer (N). Cela est problématique dans le cadre de l'identification de réseau génétique car n est très grand et il y a donc beaucoup de paramètres à estimer. Dans le chapitre suivant, nous proposerons une méthode permettant d'identifier la dynamique d'un système basée sur l'opérateur de Koopman. Lorsque celle-ci est utilisée selon la variante duale, elle permet de déterminer la dynamique sous forme d'équations différentielles lorsque $K \leq N$. Cette stratégie d'identification de réseau sera ensuite testée sur des données synthétiques et réelles.

1.4 Évaluation du modèle

De manière évidente, nous ne pouvons tester une méthode que pour des systèmes connus afin de pouvoir comparer les résultats à la réalité. Dans le contexte de la biologie, l'accès aux réseaux réels n'est généralement pas possible et les méthodes ne peuvent donc être évaluées que via des données synthétisées. Dans cette section, nous allons définir des critères pour évaluer les modèles.

La première méthode d'évaluation est basique et peut s'appliquer au cas où l'on évalue un ensemble de paramètres. Par exemple lors de l'estimation des θ_i dans le modèle d'équations différentielles et lors de régressions paramétrées pour construire un graphe. Pour de tels modèles et lorsque nous avons l'expression réelle du réseau sous la bonne forme, nous pouvons calculer directement l'erreur quadratique moyenne (Root Mean Square Error)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\theta_i - \hat{\theta}_i)^2}$$

où $\hat{\theta}_i$ est l'estimation du paramètre θ_i. L'erreur quadratique moyenne est un outil puissant qui permet de calculer les performances d'un modèle, mais elle n'est applicable que dans un nombre restreint de situations puisqu'elle nécessite l'inférence de paramètre ainsi que l'expression du réseau réel sous la bonne forme.

Les deux autres méthodes sont propres au modèle de graphe avec des arrêtes pondérées (1.2) et repose sur l'influence du seuil. Elles sont proposées, entre autres, dans les articles [12], [23] et [25].

La première se base sur le rapport entre le nombres d'arrêtes correctement identifiées et celles mal identifiées. Cette stratégie nécessite de classer les liens (ou leur absence) entre chaque nœud deux à deux en 4 catégories :

- **VP** (vrai positif) : estimation d'un lien correct ;
- **VN** (vrai négatif) : estimation de l'absence d'un lien correct ;
- **FN** (faux négatif) : oubli d'un lien ;
- **FP** (faux positif) : estimation d'un lien en trop.

Notons que ce classement dépend du seuil fixé.

Sur base de ces différentes quantités, nous définissons le taux des liens bien identifiés, le taux de vrais positifs (TVP) et le taux des faux positifs (TFP)

$$TVP = \frac{VP}{VP + FN},$$

$$TFP = \frac{FP}{FP + VN}.$$

Les deux indices sont utilisés pour tracer la courbe ROC (Receiver Operating Characteristic), celle-ci représente le taux de vrais positifs en fonction du taux de faux positifs.

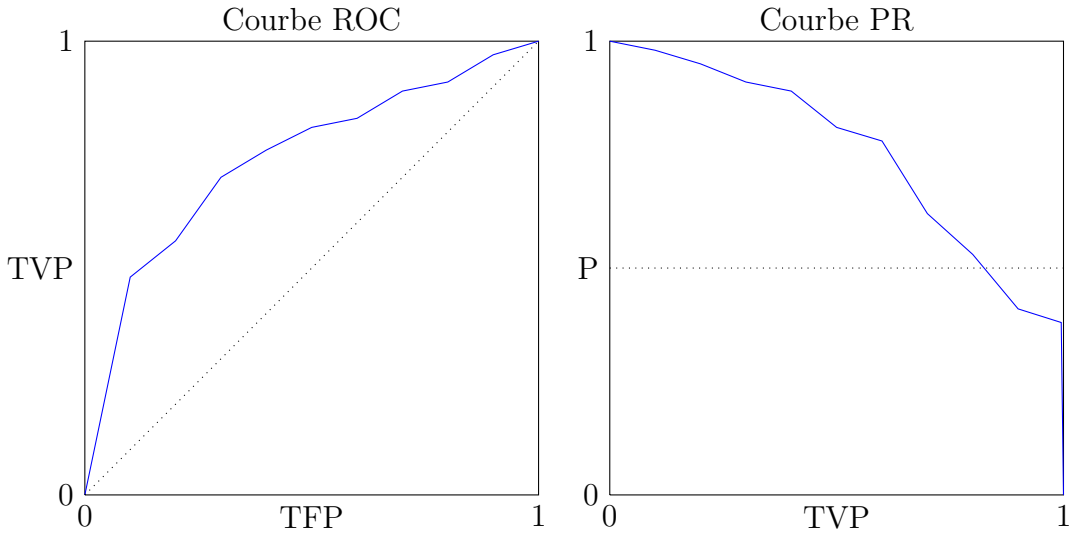


FIGURE 1.4 – Illustration des courbes ROC et PR

Un exemple se trouve à la figure 1.4, les différents taux sont obtenus en faisant varier le seuil des interactions prises en compte.

Quand le seuil est nul, aucun lien n'est identifié et donc les taux de vrais positifs et de faux négatifs sont nul. Lorsque le seuil augmente, le TVP et le TFP augmente eux aussi, mais il est souhaitable que le TVP augmente de manière plus importante que le TFP, l'objectif est donc de maximiser l'aire sous la courbe (Area Under ROC). Les performance d'une identification de réseau aléatoire sont indiquées en pointillé sur le graphique de la courbe ROC de la figure 1.4 et correspondent à une valeur AUROC de 0.5. Lors des simulations dans le chapitre 3, nous utiliserons les valeurs AUROC pour évaluer les modèles.

La dernière méthode d'évaluation présentée est la courbe de précision-rappel. Tout d'abord, la précision est définie comme la proportion de liens corrects par rapport à tous les liens identifiés, c'est-à-dire

$$P = \frac{VP}{VP + FP}.$$

La courbe PR (Precision Recall) représente la précision P en fonction de TVP. Une courbe PR est représentée à la figure 1.4. Elle peut se lire de la manière suivante : au départ le seuil est maximal, dans ce cas, il n'y a pas de faux positif (FP) et donc la précision vaut 1. Ensuite, le seuil diminue petit à petit et donc des faux positifs sont introduits. Par conséquent la précision diminue mais TVP augmente puisque le nombre de faux négatif diminue. Dans le cas idéal, nous voudrions que la précision soit toujours de 1 et donc l'objectif est aussi de maximiser l'aire sous la courbe PR. Lors d'une identification aléatoire, la précision est d'environ 0.5 peut importe le seuil.

CHAPITRE 2
MÉTHODOLOGIE

La construction de réseaux de régulation génétique vise à déterminer comment les différents gènes d'un même corps s'influencent entre eux. Pour construire de tels modèles, les puces à ADN permettent de déterminer le taux d'ARNm au sein d'un gène. Nous utiliserons l'hypothèse simplificatrice que les gènes interagissent directement entre eux au travers de l'ARNm.

La méthodologie utilisée vise à construire le réseau sous la forme d'un graphe dépendant d'un seuil et d'autre(s) paramètre(s) à fixer. Pour représenter le graphe, nous utiliserons la matrice d'adjacence définie comme en (1.1), l'objectif est donc de déterminer les différents coefficients a_{ij}.

Pour cela, nous utiliserons une stratégie en deux étapes. Premièrement, nous calculerons la variabilité du taux d'ARNm. Nous utiliserons une nouvelle méthode permettant de calculer le champ de vecteurs à l'aide de l'opérateur de Koopman et détaillée à la section 2.1. Nous présenterons ces avantages et l'illustrerons sur un exemple d'un système non linéaire. À titre de comparaison, nous calculerons également le champ de vecteur via un schéma de discrétisation basé sur les différences centrées. La seconde étape consiste à effectuer une régression qui permettra d'exprimer le champ de vecteurs en fonction des différents taux d'ARNm contenu dans chaque gène. Deux stratégies de régression seront utilisées et celles-ci sont décrites à la section 2.2. Finalement, un récapitulatif de la méthodologie utilisée se trouve à la section 2.3

2.1 Calcul du champ de vecteurs via l'opérateur de Koopman

L'opérateur de Koopman permet d'identifier un réseau sous forme d'équations différentielles mais nous ne l'utiliserons plus spécifiquement pour calculer le champ de vecteurs. Cette nouvelle technique d'identification est basée sur le principe suivant : la dynamique du système n'est pas identifiée au sein de l'espace d'état, mais au travers d'un espace infini de fonctions observables. Cette méthode présente plusieurs avantages. Tout d'abord, elle est adaptée aux petits échantillons de données. La variante duale per-

met de construire le système lorsque le nombre de données est plus petit que le nombre de paramètres à estimer. Aussi, les méthodes utilisées sont linéaires et par conséquent faciles à implémenter et rapides. L'ensemble de cette section se base sur les articles [21] et [22].

2.1.1 Définition du problème

Nommons x le vecteur réel de dimension n où chaque composante x^i représente le $i-$ème élément du réseau étudié, dans notre cas, x^i correspond au taux d'ARNm du gène i. Le système est modélisé par l'équation différentielle

$$\dot{x} = F(x) \qquad\qquad x \in X, \tag{2.1}$$

où X est l'espace d'état, dans notre cas, $X = \mathbb{R}^n$.

L'objectif de la méthode est de déterminer l'expression du champ de vecteurs $F(x)$. Nous supposons qu'il est de la forme

$$F(x) = \sum_{k=1}^{N_F} w_k h_k(x), \tag{2.2}$$

avec $\{h_k\}_{k=1}^{N_F}$ les fonctions de référence connues et arbitraires et $\{w_k = (w_k^1, ..., w_k^n)^T\}_{k=1}^{N_F}$ les coefficients à identifier.

Pour utiliser la méthode, les données doivent être de la forme de K paires (x_k, y_k) définies comme suit :

$$x_k = \bar{x}_k + \epsilon(x_k) \text{ et } y_k = \bar{y}_k + \epsilon(y_k), \tag{2.3}$$

où ϵ désigne le bruit. Nous supposons que le bruit est gaussien de moyenne nulle et de variance σ. Considérons $\varphi^t(x_0)$ le flot du système (2.1), c'est-à-dire que $\varphi^t(x_0)$ est la solution du système pour la condition initiale x_0. Les mesures doivent vérifier que

$$\bar{y}_k = \varphi^{T_s}(\bar{x}_k), \tag{2.4}$$

où T_s est la période d'échantillonnage identique pour toutes les paires. Nous notons les observations x_k^i et nous utiliserons la conventions suivante, l'indice supérieur représente la composante avec $i = 1, ..., n$ et l'indice inférieur représente le numéro de l'observation avec $k = 1, ...K$.

En pratique, les données issues des puces à ADN ne sont pas tout à fait de cette forme, mais cela sera discuté dans la section 2.1.5.

2.1.2 Définition et propriétés de l'opérateur de Koopman

L'opérateur de Koopman permet de changer d'espace mais de garder la dynamique du système et l'identification au sein de ce nouvel espace sera linéaire et donc plus aisée. Pour cela, considérons l'espace de fonctions

$$\mathcal{F} = \{f : X \to \mathbb{R} | f \text{ observable}\},$$

et définissons l'opérateur de Koopman U^t sur cet espace $\mathcal{F}$.

Définition 1 *L'opérateur de Koopman U^t est l'application*

$$\begin{aligned} U^t : \mathcal{F} &\longrightarrow \mathcal{F} \\ f &\longmapsto U^t(f) = f \circ \varphi^t. \end{aligned}$$

où $\mathcal{F}$ est un espace fonctionnel arbitraire et φ^t le flot du système.

Cette opérateur est linéaire, en effet $\forall \alpha, \beta \in \mathbb{R}, \forall f, g \in \mathcal{F}$:

$$\begin{aligned} U^t(\alpha f + \beta g) &= (\alpha f + \beta g) \circ \varphi^t \\ &= \alpha f \circ \varphi^t + \beta g \circ \varphi^t \\ &= \alpha U^t(f) + \beta U^t(g). \end{aligned}$$

Cela constitue un des avantages majeurs de la méthode d'identification de réseau. L'ensemble des opérateurs $\{U^t\}_{t\geq 0}$ forment un C_0-semi-groupe d'opérateurs. Le générateur infinitésimal L est défini par

$$L = F \cdot \nabla \tag{2.5}$$

où ∇ désigne le gradient qui s'exprime par

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, ..., \frac{\partial f(x)}{\partial x_n} \right).$$

Nous utiliserons la notation suivante, néanmoins L n'est pas borné

$$U^t = e^{Lt}. \tag{2.6}$$

Si les fonctions f sont continûment différentiables, nous pouvons alors exprimer leur dynamique

$$\frac{\partial f \circ \varphi^t}{\partial t} = (F \cdot \nabla) f \tag{2.7}$$

où ∇ désigne la gradient et $f \in \mathcal{F}$. Nous retrouvons dans l'équation l'expression de L (2.5). Le flot de ce système est donné par l'opérateur de Koopman U^t Nous noterons

$$\dot{f} = \frac{\partial f \circ \varphi^t}{\partial t}, \tag{2.8}$$

ainsi l'équation (2.7) devient

$$\dot{f} = Lf. \tag{2.9}$$

Autrement dit, nous avons associé au système (2.9) une nouvelle expression dans un espace de dimension infinie mais linéaire. La linéarité de cet espace permet de rendre l'identification de la dynamique plus facile.

2.1.3 Description de la méthode principale

Désormais, nous avons les bases nécessaires pour expliquer les trois étapes de la méthode comme illustré à la figure 2.1. L'objectif est d'identifier le champ de vecteur F défini par (2.1). Pour cela nous commençons par transférer les données dans l'espace fonctionnel $\mathcal{F}$. Ensuite, dans cet espace, nous identifions l'opérateur de Koopman U^{T_s} et nous obtenons l'expression du générateur infinitésimal L grâce à l'équation (2.6) qui les lie. Ce générateur représente la dynamique du système et permet d'obtenir F via la définition de L (2.5).

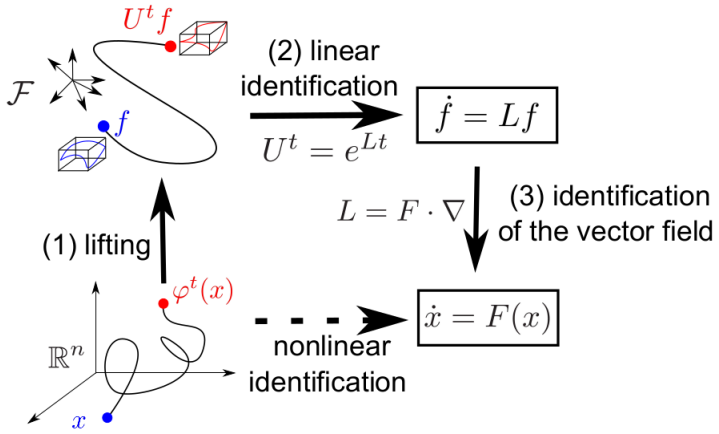


FIGURE 2.1 – Illustration des 3 étapes pour identifier le champ de vecteurs. Source : [21]

2.1.3.1 Transfert des données

Les données mises à disposition sont dans l'espace $X = \mathbb{R}^n$ et nous allons les "transférer" dans l'espace $\mathcal{F}$. Comme cet espace est de dimension infinie, nous utilisons un sous-espace $\mathcal{F}_N \subset \mathcal{F}$. Ce dernier est engendré par N fonctions linéairement indépendantes $\{g_k\}_{k=1}^N$. Ces fonctions sont arbitraires et nous choisissons l'ensemble des monômes p_k dont le degré est inférieur ou égal à m. Dans ce cas, le nombre de paramètres à inférer est égal au nombre de fonction de base et est donné par

$$N = \frac{(n+m)!}{n!m!}. \tag{2.10}$$

Nous nommons $p(x) = (p_1(x), ..., p_N(x))^T$, le vecteur reprenant les N polynômes de la base. À partir des paires de données $(x_y, y_k) \in \mathbb{R}^{n\times 2}$, nous allons construire de nouvelles paires $(p(x_k), p(y_k)) \in \mathbb{R}^{N\times 2}$. Nous construisons alors deux matrices P_x et P_y de dimensions $K \times N$ définies par

$$P_x = \begin{pmatrix} p(x_1)^T \\ \vdots \\ p(x_K)^T \end{pmatrix} \text{ et } P_y = \begin{pmatrix} p(y_1)^T \\ \vdots \\ p(y_K)^T \end{pmatrix}. \tag{2.11}$$

Intéressons-nous au lien entre x_k ey y_k.

$$p(y_k) \overset{(2.3)}{=} p(\bar{y}_k + \epsilon(y_k)) \overset{(2.4)}{=} p(\varphi^{T_s}\bar{x}_k + \epsilon(y_k)) \overset{(2.3)}{=} p(\varphi^{T_s}(x_k - \epsilon(x_k)) + \epsilon(y_k)).$$

Nous pouvons alors approximer $p(y_k)$ par

$$p(y_k) \approx U^{T_s} p(x_k) + \mathcal{O}(||\epsilon||). \tag{2.12}$$

Remarquons que cette approximation est valable pour d'autres fonctions de base que les polynômes p_k.

2.1.3.2 Identification de l'opérateur de Koopman

Nous allons identifier l'opérateur de Koopman U^t avec $t = T_s$ la période d'échantillonnage des données. Comme cet opérateur est de dimension infinie, en pratique, nous utiliserons U_N, la projection sur l'espace fini $\mathcal{F}_N$. Formellement, nous définissons U_N comme

$$U_N = P_N U^{T_s}|_{\mathcal{F}_N}$$

où $U^{T_s}|_{\mathcal{F}_N}$ est la restriction de l'opérateur à l'espace $\mathcal{F}_N$ et P_N l'opérateur de projection orthogonale dans l'espace $\mathcal{F}_N$.

Considérons les fonctions f et $U_N f$ avec $f \in \mathcal{F}_N$, nous pouvons les exprimer à l'aide de la base $\{p_k\}_{k=1}^N$, c'est-à-dire qu'il existe a et b tel que

$$f = a^T p \text{ et } U_N f = b^T p. \tag{2.13}$$

Nous définissons $\bar{U}_N \in \mathbb{R}^{N\times N}$ la représentation matricielle de l'opérateur U_N, comme l'opérateur de Koopman est linéaire, il vérifie

$$\bar{U}_N a = b. \tag{2.14}$$

En combinant (2.13) et (2.14), nous obtenons

$$U_N f = b^T p = (\bar{U}_N a)^T p. \tag{2.15}$$

Cette égalité est valide en particulier pour $f = p_i$ avec $i \in \{1, ..., N\}$, nous obtenons donc

$$U_N f = U_N p_i \overset{(2.15)}{=} (\bar{U}_N e_i)^T p = c_i p$$

où c_i désigne la $i-$ème colonne de $\bar{U}_N$. Cela signifie que chaque colonne de $\bar{U}_N$ est obtenue en réalisant la projection orthogonale de l'image de p_i par U^{T_s} dans le sous-espace $\mathcal{F}_N$.

Dès lors, pour pouvoir estimer $\bar{U}_N$, nous utiliserons l'expression de l'opérateur de projection orthogonale sur $\mathcal{F}_N = span\{p_1, ...p_N\}$ données par

$$P_N m = \underset{\tilde{m}\in span\{p_1,...p_N\}}{\arg\min} \sum_{k=1}^{K} |\tilde{m}(x_k) - m(x_k).|^2,$$

Pour pouvoir résoudre ce problème, il faut que $K \geq N$. Dans le cas où cette condition ne peut être vérifié, alors il est possible d'utiliser la méthode duale qui nécessite justement que $K \leq N$. Via la solution des équations des moindres carrés, nous obtenons

$$P_N m = p^T P_x^\dagger \begin{pmatrix} m(x_1) \\ \vdots \\ m(x_K) \end{pmatrix}$$

où $P_x^\dagger$ est le pseudo-inverse de P_x.

Maintenant, nous pouvons trouver l'approximation de l'opérateur de Koopman. Pour cela, développons l'expression pour $m = U^{T_s} p_i$,

$$P_N(U^{T_s} p_i) = p^T P_x^\dagger \begin{pmatrix} U^{T_s} p_i(x_1) \\ \vdots \\ U^{T_s} p_i(x_n) \end{pmatrix} \overset{(2.12)}{\approx} p^T P_x^\dagger \begin{pmatrix} p_i(y_1) \\ \vdots \\ p_i(y_n) \end{pmatrix}.$$

Ce résultat est valable $\forall i = 1, ..., N$ et donc finalement, nous obtenons

$$P_N U^{T_s} p^T = U_N p^T = p^T P_x^\dagger P_y.$$

L'expression matricielle $\bar{U}_N$ vaut donc

$$\bar{U}_N \approx P_x^\dagger P_y.$$

Via (2.6), nous calculons $\bar{L}_{data}$ qui est l'approximation de $\bar{L}_N$ la représentation matricielle de L_N avec $L_N = P_N L|_{\mathcal{F}_N}$. Nous obtenons

$$\bar{L}_{data} = \frac{1}{T_s} \log(P_x^\dagger P_y). \tag{2.16}$$

2.1.3.3 Identification du champ de vecteurs

Finalement, il reste à identifier le champ de vecteurs F, c'est-à-dire les coefficients $w_k = (w_k^1, ..., w_k^n)$ définis par (2.2). Pour cela, nous utilisons l'approximation $\bar{L}_{data}$ obtenue à la section précédente, ainsi que la relation (2.5) qui lie L et le champ de vecteur.

Désignons par p_l le polynôme de la base $\{p_k\}_{k=1}^N$ tel que $p_l(x) = x_j$ et où x_j représente la $j - ème$ composante du vecteur x. Dans ce cas,

$$L_N p_l = P_N(F \cdot \nabla) p_l = P_N(F \cdot \nabla p_l) = P_N F_j = F_j.$$

Nous exprimons $L_n p_l$ comme $p^T(\bar{L}_n e_l)$, dès lors

$$F_j = p^T(\bar{L}_N e_l) \approx p^T(\bar{L}_{data} e_l).$$

Autrement dit, la colonne l de $\bar{L}_{data}$ contient la décomposition de F_j dans la base des polynômes, c'est-à-dire les w_k^j et donc

$$\hat{w}_k^j = \left(\bar{L}_{data}\right)_{kl}.$$

Nous utiliserons plus particulièrement la méthode lifting pour estimer le champ de vecteurs au différents points x_k. Ce champ de vecteurs est donné par

$$\begin{pmatrix} \hat{F}_j(x_1) \\ \vdots \\ \hat{F}_j(x_K) \end{pmatrix} = P_X(\bar{L}_{data} e_l)$$

où l désigne l'indice du polynôme tel que $p_l(x) = x_j$.

2.1.4 Description de la méthode duale

Lors de la deuxième étape, pour calculer les solutions des équations de moindre carré, nous avons supposé que $K \geq N$. C'est-à-dire que nous pouvons utiliser la méthode principale uniquement lorsque le nombre d'observations est plus grand que la dimension de l'espace $\mathcal{F}_N$. Nous allons présenter une méthode duale qui pourra être utilisée lorsque $K \leq N$. Celle-ci suit le même schéma que la première avec 3 étapes.

2.1.4.1 Transfert des données

La première étape est similaire au transfert de données réalisé à la section 2.1.3.1, néanmoins, nous ne considérons plus la même base de fonctions $\{g_k\}_{k=1}^N$. À la place des polynômes, nous utilisons des fonctions Gaussiennes de la forme

$$g_k(x) = e^{-\gamma||x-x_k||^2}, \tag{2.17}$$

avec $\gamma > 0$ un paramètre. Nous définissons alors les matrices P_x et P_y,

$$P_x = \begin{pmatrix} g(x_1)^T \\ \vdots \\ g(x_K)^T \end{pmatrix} \text{ et } P_y = \begin{pmatrix} g(y_1)^T \\ \vdots \\ g(y_K)^T \end{pmatrix}.$$

2.1.4.2 Identification de l'opérateur de Koopman

Nous commençons par définir la matrice $\tilde{U}_K$ de taille $K \times K$ qui correspond à la représentation matricielle de U^{T_s} dans l'espace engendré par $f(x_k), f \in \mathcal{F}$. Nous la définissons à partir de $\bar{U}_N$ et de P_x qui peut être vu comme un changement de coordonnées,

$$\tilde{U}_K \approx P_x \bar{U}_N P_x^\dagger = P_y P_x^\dagger. \tag{2.18}$$

Dans ce cas, $\forall f \in \mathcal{F}_N$,

$$\begin{pmatrix} U^{T_s} f(x_1) \\ \vdots \\ U^{T_s} f(x_K) \end{pmatrix} \approx \tilde{U}_K \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_K) \end{pmatrix}. \tag{2.19}$$

L'approximation du générateur infinitésimal est à nouveau obtenue via (2.6), son expression est

$$\tilde{L}_{data} = \frac{1}{T_s} \log(P_y P_x^\dagger). \tag{2.20}$$

Montrons maintenant que cette méthode nécessite $N \geq K$. Nous désignons la $i-$ème ligne de $\tilde{U}_K$ par l_i, elle représente les coefficients de $U^{T_s} f(x_i)$ dans la base des $\{f(x_i)\}_{i=1}^K$. Considérons l'équation (2.19) avec $f(x) = g_k(x)$, alors

$$\begin{pmatrix} U^{T_s} g_k(x_1) \\ \vdots \\ U^{T_s} g_k(x_K) \end{pmatrix} \overset{(2.12)}{\approx} \begin{pmatrix} g_k(y_1) \\ \vdots \\ g_k(y_K) \end{pmatrix} = \tilde{U}_K \begin{pmatrix} g_k(x_1) \\ \vdots \\ g_k(x_K) \end{pmatrix}.$$

Nous pouvons écrire $g_k(y_i) \approx l_i g_k(x)$, cette approximation avec valable $\forall k = 1, ..., N$, donc

$$(g_1(y_i), ..., g_N(y_i)) \approx l_i P_x.$$

Le vecteur r_i représente les inconnues, elles sont donc au nombre de K, tandis que les valeurs $g_1(y_i), ..., g_N(y_i)$ représentent les données, il faut donc que $K \leq N$. En pratique, au vu du choix des fonctions g_k, N est fixé à K.

2.1.4.3 Identification du champ de vecteurs

Pour identifier le champ de vecteurs F, nous utilisons le champ de vecteurs évalué aux points $\{x_k\}_{k=1}^K$. Il faut ensuite résoudre n problèmes de régression pour obtenir les coefficients $\{w_k = (w_k^1, ..., w_k^n)\}_{k=1}^{N_F}$.

Commençons par approximer le champ de vecteurs F évalué en x_k. Dans la section précédente, nous avons donc obtenu une approximation de la représentation matricielle L_N notée $\tilde{L}_{data}$. Ainsi, nous avons

$$\begin{pmatrix} F(x_1) \cdot \nabla f(x_1) \\ \vdots \\ F(x_K) \cdot \nabla f(x_K) \end{pmatrix} \overset{(2.5)}{=} \begin{pmatrix} Lf(x_1) \\ \vdots \\ Lf(x_K) \end{pmatrix} \approx \tilde{L}_{data} \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_K) \end{pmatrix}$$

où $f \in \mathcal{F}$. Le choix de f est arbitraire, nous choisissons $f(x) = x$ et dans ce cas nous obtenons l'approximation du champ de vecteurs $\hat{F}$,

$$\begin{pmatrix} \hat{F}(x_1)^T \\ \vdots \\ \hat{F}(x_K)^T \end{pmatrix} \approx \tilde{L}_{data} \begin{pmatrix} x_1^T \\ \vdots \\ x_K^T \end{pmatrix}. \tag{2.21}$$

Rappelons que via la définition du système initial (2.1) le vecteur $F(x_k)$ est de dimension n.

Finalement, les coefficients w_l^j définis par (2.2) peuvent être estimé à l'aide de l'expression du champ de vecteurs évalué aux différents points de notre échantillon. Pour cela, il faut résoudre

$$\hat{F}_j(x_k) = \sum_{l=1}^{N_F} \hat{w}_l^j h_l(x_k)$$

où $j = 1, ..., n$ et $k = 1, ..., K$. Nous écrivons le problème sous forme matricielle

$$\begin{pmatrix} \hat{F}_j(x_1) \\ \vdots \\ \hat{F}_j(x_K) \end{pmatrix} = \underbrace{\begin{pmatrix} h(x_1)^T \\ \vdots \\ h(x_K)^T \end{pmatrix}}_{\overset{not}{=} H_x} \begin{pmatrix} \hat{w}_1^j \\ \vdots \\ \hat{w}_{N_F}^j \end{pmatrix} \tag{2.22}$$

où $h(x) = (h_1(x), ..., h_{N_F}(x))$. Pour obtenir les $n \times N_F$ coefficients w_l^j, il faut résoudre le problème de régression (2.22) pour $j = 1, ..., n$. Cette régression peut être améliorée en

utilisant une contrainte sur la norme des coefficients w_l^j. Cela peut se faire par exemple grâce à la méthode lasso développée à la section 2.2.2. Notons que la méthode n'impose pas de forme pour les fonctions h.

2.1.5 Discussion

Nous avons présenté deux méthodes basées sur l'opérateur de Koopman qui permettent d'identifier un réseau sous la forme d'un système dynamique. Ces méthodes contiennent des approximations, notamment lorsqu'on projette l'opérateur de dimension infinie dans un sous-espace de dimension finie, mais l'article [21] propose un résultat de convergence.

Dans les conditions définies dans les descriptions des deux méthodes, chacune des variantes vérifie pour des données non bruitées que

$$\lim_{N\to\infty} \lim_{K\to\infty} \lim_{T_s\to 0} \left|\left|\hat{F}_j - F_j\right|\right| = 0$$

avec une probabilité de 1. De plus pour la stratégie principale, si nous considérons $h_k = p_k$, alors $\forall N \geq N_F$,

$$\lim_{T_s\to 0} \hat{w}_k^j = w_k^j, \qquad \forall k = 1, ..., N_F$$

avec une probabilité de 1. Ce résultat assure donc l'efficacité de la méthode dans la cadre idéal, mais dans la pratique, les données sont bruitées et le nombre de données K est limité.

Remarquons également que la méthode suppose un format de données bien précis tel que (2.4) est vérifié. Les données sont régulièrement sous la forme de séries temporelles avec la même période d'échantillonnage T_s. Considérons $\{z_k = (z_k^1, ..., z_k^n)\}_{k=1}^{\tilde{K}}$ les mesures temporelles de chaque variable, dans ce cas, les paires (x_y, y_k) utilisées pour la méthode duale sont

$$(x_k, y_k) = (z_k, z_{k+1}) \qquad \forall k = 1, ..., \tilde{K} - 1.$$

Le nombre de paires de données K est alors donné par $\tilde{K} - 1$.

Par la suite, nous nommerons *lifting* la méthode de calcul du champs de vecteurs basée sur l'opérateur de Koopman. Dans la pratique, nous utiliserons la méthode duale pour calculer le champ de vecteurs aux différents points x_k via l'équation (2.21). Cette méthode nécessite d'utiliser les fonctions g_k définies en (2.17) qui dépendent d'un paramètre γ, nous discuterons de l'influence de ce paramètre lors de l'évaluation du modèle.

2.1.6 Illustration

Dans cette section, nous présenterons brièvement un exemple pour montrer l'efficacité de la méthode lifting. Pour cela, nous utilisons un système non linéaire généré

aléatoirement repris de l'article [21]. Chaque composante du système est influencée par $n_{inter.}$ monômes de degré 2 et 3. Ce système est de la forme

$$\dot{x_j} = w_1^j x^j + \sum_{k=2}^{N_F} w_k^j h_k \qquad \forall j = 1, ...n, \tag{2.23}$$

où les fonctions h_k sont des monôme de la forme $(x^m)^p (x^l)^q$ où p et q sont tels que $p + q$ vaut 2 ou 3 et $m, l \in \{1, .., n\}$. Un grand nombre des coefficients w_k^j est nul et seuls $n_{inter.}$ coefficients w_k^j sont non nuls. Les paramètres suivants sont fixés : $n = 30$, $n_{inter.} = 5$ et $K = 200$.

Nous allons estimer $F(x_k)$ le champ de vecteurs calculés pour les différents points donnés. Il faut choisir entre la variante principale et la variante duale. Dans le cas où nous utiliserions la variante principale, il faudrait que le degré des monômes des fonctions de base m soit au moins de degré 3 au vu de la forme du système. Dans ce cas, le nombre de paramètres à inférer serait donné par (2.10) et vaudrait

$$N = \frac{(n+m)!}{n!m!} = \frac{33!}{30!3!} = 5456.$$

Ce n'est donc pas possible d'utiliser la variante principale pour ce choix de m, sinon $K \geq N$. Nous utiliserons alors la méthode duale. Celle-ci nécessite de fixer la paramètre γ.

Pour analyser les performance de la méthode, nous utiliserons l'erreur quadratique moyenne. En effet, comme nous avons accès à la dynamique du système, nous pouvons connaître la valeur exacte de $F(x_k)$. L'erreur quadratique moyenne est calculée par

$$RMSE = \frac{\|\hat{F}(X) - F(X)\|}{\sqrt{n.K}}.$$

Cette erreur est ensuite normée en la divisant par la valeur moyenne des valeurs absolues des composantes de $F(X)$.

Pour générer les données, une intégration numérique est utilisée à partir de conditions initiales aléatoires. Un test est également effectué en ajoutant un bruit Gaussien de moyenne nulle et de variance $\sigma = 0.01$. L'erreur obtenue en fonction de T_s se trouve à la table 2.1.

Ces résultats attestent de l'efficacité de la méthode lifting pour identifier un champ de vecteurs d'un système non linéaire. Nous pouvons constater l'influence du pas de temps T_s : au plus il est petit, au plus l'approximation du champ de vecteurs est correcte. Néanmoins dans le cas des données bruitées, un pas trop petit amène à des erreurs plus importantes. L'erreur pour le cas non bruité se situe entre 0.004 et 0.064 en fonction du pas. Lorsque les données sont bruitées, l'erreur est plus importante, mais pour un bon choix de pas de temps, l'erreur est de l'ordre de 0.1. Malheureusement, en pratique, nous n'aurons pas toujours la possibilité de pouvoir réduire le pas de temps.

T_s	0.1	0.02	0.03	0.04	0.05
$\sigma = 0$	0.004	0.013	0.026	0.044	0.064
$\sigma = 0.01$	0.301	0.167	0.122	0.099	0.112

TABLE 2.1 – NRMSE obtenus pour l'estimation du champ de vecteurs d'un système de la forme (2.23)

Nous sommes donc parvenus à identifier le champ de vecteurs du réseau (2.23) avec la méthode lifting duale. Si nous avions utilisé une méthode plus classique, il aurait fallu estimer les N_F paramètres du modèle (2.23), ce problème est largement sous-déterminé. La méthode lifting est donc une bonne alternative lorsque le nombre d'observations est faible.

2.2 Régression

Nous avons montré comment obtenir le champ de vecteurs via l'opérateur de Koopman. En pratique, nous utiliserons la variante duale. Voyons maintenant comment déterminer les influences entre les éléments. Les données du problème sont les suivantes : X représente les données et $\hat{F}$ l'estimation du champ de vecteur. Elles sont définies comme suit

$$X = \begin{pmatrix} x_1^1 & \dots & x_1^n \\ \vdots & \ddots & \vdots \\ x_K^1 & \dots & x_K^n \end{pmatrix} \overset{not}{=} \begin{pmatrix} x_1 \\ \vdots \\ x_K \end{pmatrix},$$

$$\hat{F} = \begin{pmatrix} \hat{F}_1(x_1) & \dots & \hat{F}_n(x_1) \\ \vdots & \ddots & \vdots \\ \hat{F}_1(x_K) & \dots & \hat{F}_n(x_K) \end{pmatrix} \overset{not}{=} \begin{pmatrix} | & & | \\ \hat{F}_1 & \dots & \hat{F}_n \\ | & & | \end{pmatrix}. \tag{2.24}$$

L'objectif du problème de régression est de déterminer une fonction R tel que

$$\hat{F} = R(X). \tag{2.25}$$

En fonction du choix de la régression, R peut prendre de nombreuses formes. Selon la forme de R, nous pourrons définir les interactions entre les différents éléments et construire la matrice d'adjacence. En effet, la fonction R explique le comportement de $\hat{F}$, le champ de vecteurs c'est-à-dire la variabilité des x_i, en fonction des x_i.

Nous utiliserons deux types de régression, la première méthode est basée sur les arbres décisionnels où R est une fonction constante par morceau. La seconde est une régression linéaire classique mais avec une pénalité imposant une condition de spartité sur les coefficients.

2.2.1 Régression via les arbres décisionnels

La régression via les arbres de décision vise à sectionner l'espace d'échantillonnage $\mathbb{R}^n$ selon des critères binaires. Pour chacune des régions ainsi définies, une valeur de sortie est attribuée. L'explication de cette méthode est basée sur les références [6], [8], [9] et [15] et les codes utilisés proviennent de [7].

L'algorithme est représenté à la figure 2.2 et se présente comme suit : les différentes observation $x_1, ..., x_K$ sont séparées en deux selon un critère du type $x^i < s$ où i et s doivent être fixés. Ensuite, le procédé de séparation est reproduit sur chacun des deux sous-ensembles d'observations jusqu'à ce que le nombre d'éléments dans chaque région soit inférieur ou égal à n_{min}. Une fois l'espace $\mathbb{R}^n$ partitionné en région, une valeur de sortie, notée t_i est associée à chacune. La fonction R est alors définie par

$$Y \approx R(X) = \sum_{m=1}^{M} t_i \, \mathcal{I}_{r_m}(X) \tag{2.26}$$

où $\mathcal{I}_{r_m}$ est l'indicatrice associé à la région r_m et M le nombre de régions.

Il existe plusieurs algorithmes pour obtenir un critère de scission, celui utilisé vise à réduire la variance des sorties correspondant aux observations d'une même région. Il fonctionne de la manière suivante.

1. Choix aléatoire de P variables x^i parmi n.
2. Choix aléatoire d'une valeur de scission s pour chacune des P variables. Cette valeur est choisie à l'intérieur de l'intervalle $[x^i_{min}, x^i_{max}]$.
3. Sélection du meilleur critère s^* selon une fonction score S, c'est-à-dire

$$s^* = \arg \max_{m=1,..,P} S(m, y).$$

Considérons le m-ème critère et nommons E l'ensemble d'observations considéré et E_1 et E_2 les deux sous-ensembles formés selon le critère $x^i < s$. La fonction score est définie par

$$S(m, y) = \frac{Var(y|E) - \frac{|E_1|}{|E|} Var(y|E_1) - \frac{|E_2|}{|E|} Var(y|E_2)}{Var(y|E)},$$

où $Var(y|E)$ désigne la variance des sorties liées aux observations de E et $|E|$ le nombre d'observations dans l'ensemble E. Autrement dit, le critère de sélection est choisi pour que les valeurs de sorties des observations d'une même région soient le plus proches possible.

Bien que l'arbre de décision optimise la fonction score, il contient une part d'aléatoire au vu des critères de scission. Pour améliorer les performances, il est courant d'utiliser une forêt, c'est-à-dire un ensemble d'arbres D. Dans ce cas, l'algorithme construit D arbres indépendants et la fonction de régression est définie grâce à une moyenne des arbres

$$Y \approx R(X) = \frac{1}{D} \sum_{j=1}^{D} R_j(x),$$

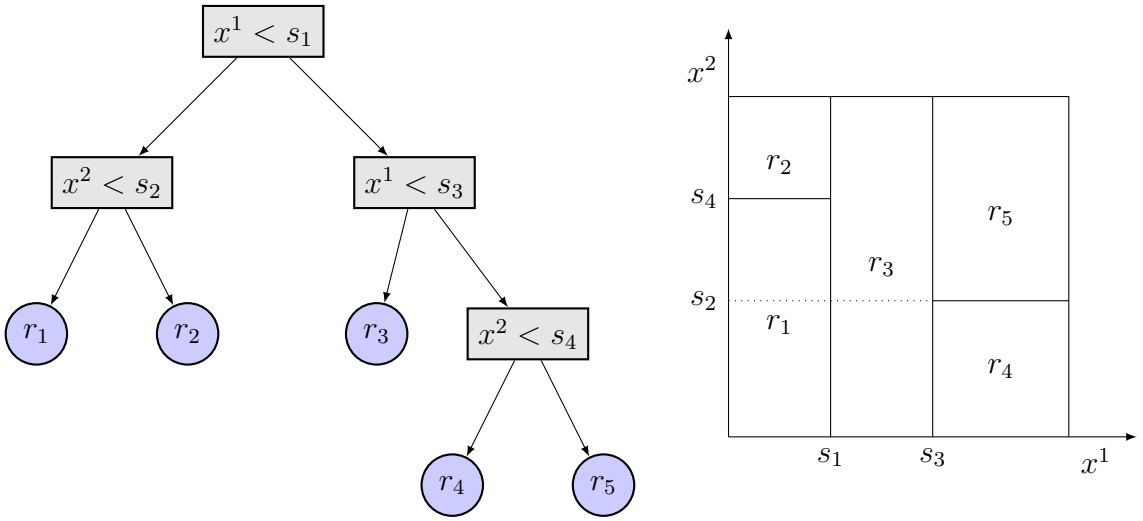


FIGURE 2.2 – Régression via les arbres décisionnels

où R_j est la fonction de régression associée à l'arbre j.

La méthode repose donc sur 3 paramètres : P, n_{min} et D. Discutons de l'influence de ceux-ci. Le premier permet de tester plusieurs critères de scission, au plus P est petit, au plus la construction de l'arbre est aléatoire. Dans le cas où $P = 1$ l'arbre est complètement aléatoire. Lorsque P vaut n alors l'aléatoire ne porte que sur le choix de la valeur de scission. Dans la plupart des problèmes de régression lorsque P augmente, l'erreur diminue, c'est pourquoi nous utiliserons $P = n$.

Concernant le choix de n_{min} l'article [6] suggère que le choix le plus robuste est $n_{min} = 5$, mais notons que l'analyse a été réalisée sur des jeux données où le nombre de données K est entre 300 et 8000. La taille des jeux de données utilisés dans les chapitres 3 et 4 sont de taille plus petites (entre 12 et 210), nous verrons comment adapter ce paramètre.

Finalement, il reste à discuter du choix du paramètre D représentant le nombre d'arbres utilisés pour réalisé la régression. Au plus D est important, au plus l'erreur diminue. Néanmoins augmenter D nécessite des calculs supplémentaires et il faut donc trouver un compromis.

L'avantage premier de cette méthode de régression réside dans le fait que, contrairement aux autres, elle n'impose pas une forme de dynamique et est théoriquement valable pour tout type de système. L'efficacité de cet algorithme a été démontrée sur une série de jeux de données dans la référence [6]. Notons néanmoins, que, dans notre cadre, la composition des jeux de données est différente : le nombre de données K est faible vis à vis du nombre de variables n.

Nous avons donc montré comment réaliser une régression de type (2.25), voyons maintenant comment obtenir la matrice d'adjacence. Lors de la construction de l'arbre,

le nombre de fois où un attribut x^i est utilisé pour un critère de scission atteste de l'importance ou non de l'influence de cet attribut x^i sur la sortie $\hat{F}$. Cette influence est d'autant plus importante si le critère de scission permet de fortement augmenter la fonction score. C'est selon ce principe qu'est calculée la matrice d'adjacence, ainsi l'élément $\tilde{a}_{ij}$ est défini par

$$\tilde{a}_{ij} = \sum S(m, y),$$

où la somme est réalisée sur les critères tels que x^i est la variable de scission et j désigne la composante pour laquelle la régression est effectuée. À nouveau, plusieurs arbres sont utilisés et donc la mesure de l'influence est une moyenne des coefficients $\tilde{a}_{ij}$ obtenus pour chaque arbre. En général, le graphe obtenu est complètement connecté, pour obtenir un graphe moins dense et non pondéré qui mesure uniquement les influences principales, il suffit de fixer un seuil.

Dans la pratique, $\hat{F}$ est de dimension $K \times n$ où chaque colonne représente l'estimation d'une composante du champ de vecteurs comme illustré en (2.24). L'algorithme avec régression n'effectue pas en même temps la régression pour les n composantes de du champ de vecteurs, mais réalisent n régressions indépendantes pour chacune des composantes. Comme il y a n problèmes de régression indépendant, le choix des arrêtes au dessus du seuil est effectué pour chaque composante indépendamment. Autrement dit, la matrice d'adjacence non pondérée est définie par

$$a_{ij} = \begin{cases} 1 \text{ si } \tilde{a}_{ij} > \tilde{a}_{\bullet j,min} + seuil(\tilde{a}_{\bullet j,max} - \tilde{a}_{\bullet j,min}) \\ 0 \text{ sinon} \end{cases}$$

où $\tilde{a}_{\bullet j,min}$ et $\tilde{a}_{\bullet j,min}$ sont les éléments minimal et maximal de la $j - ième$ colonne de $\tilde{A}$.

2.2.2 Régression via la méthode lasso

Nous allons présenter une seconde méthode de régression qui, cette fois-ci est moins générale puisqu'elle est linéaire, mais permet de sélectionner les variables les plus pertinentes à l'aide d'une contrainte. Les références [9] et [13] ont été utilisées pour décrire la méthode lasso.

La fonction de régression est définie comme

$$\hat{F} = R(X) = X\beta + \epsilon$$

où ϵ correspond au bruit des données. Contrairement aux méthodes de régression linéaire classique, la méthode lasso ajoute une contrainte sur la norme de β pour imposer qu'une partie des coefficients soient nuls. La matrice β est obtenue en réalisant le problème d'optimisation ci dessous.

$$\hat{\beta}(\lambda) = \arg \min_{\beta \in \mathbb{R}^{n \times K}} \left(\frac{1}{2} ||\hat{F} - X\beta||_2^2 + \lambda ||\beta||_1 \right), \tag{2.27}$$

où γ est un paramètre et $\| \bullet \|_1$ et $\| \bullet \|_2$ sont les normes usuelles.

Le paramètre λ influence le nombre de coefficients nuls, lorsque $\lambda = 0$, la méthode lasso correspond à une problème de régression classique sans contrainte. Au fur et à mesure que λ augmente, la contrainte est importante et donc la norme de β doit diminuer. Le choix de la norme $\|\bullet\|_1$ permet de forcer que des coefficients soient exactement nuls. Notons que le choix de λ dépend très fortement de l'échelle utilisée et du nombre de composantes.

L'avantage premier de la méthode lasso réside dans le fait qu'elle est applicable dans le cas où $n > K$, contrairement à la régression linéaire classique. Bien que la régression soit linéaire, le choix des variables peut permettre, dans certains cas, de donner une approximation de modèles non linéaires. En effet, la méthode lasso permet de sélectionner les bonnes variables dans le cas non linéaire, bien que les coefficients ne soient pas exacts [29]. Elle permet, en plus, de sélectionner les variables les plus importantes ce qui permet de simplifier les modèles obtenus. Par contre, lorsqu'il y a de fortes corrélation entre deux variable importantes, la méthode lasso risque de n'en sélectionner qu'une seule pour expliquer le modèle.

L'unicité de la solution du problème de minimisation (2.27) n'est pas garantie lorsque le nombre de variable est plus important que le nombre d'observations [27]. Dans notre cas, le nombre de variables vaut n puisqu'il s'agit d'une régression linéaire et le nombre d'observation vaut K. Lorsque la méthode lasso sera utilisée, nous vérifierons toujours que $n \leq K$ et donc l'unicité de la solution est garantie.

Voyons maintenant comment construire la matrice d'adjacence à partir de cette régression. Les coefficients β correspondent aux éléments de la matrice d'adjacence pondérée $\tilde{A}$. Pour construire la matrice d'adjacence non pondérée, il faut fixer un seuil, or en fonction du paramètre λ, le graphe est déjà peu connecté, c'est pourquoi nous utiliserons un seuil de 0. Autrement dit, les éléments non nuls de la matrice d'adjacence correspondent aux coefficients non nuls de la régression avec lasso. Bien que le seuil reste fixé à 0, il sera tout de même possible d'évaluer la méthode à l'aide des courbes ROC en faisant varier λ.

Tout comme la régression avec les arbres de décision, dans le cadre de l'identification de réseaux de régulation génétique, nous effectuerons n problèmes de régression indépendants pour chacune des composantes du champ de vecteurs.

2.3 Récapitulatif méthodologie

La figure 2.3 récapitule la méthode utilisée. Au départ, la matrice Z contient le taux d'ARNm dans les n gènes pour $\tilde{K}$ instants donnés. Pour rappel, l'objectif est de déterminer la matrice d'adjacence A définie par (1.1) et dont l'élément a_{ij} mesure l'influence du gène i sur le gène j.

Après avoir mis les données sous la bonne forme, le champ de vecteurs $\hat{F}$ est calculé grâce à la méthode basée sur l'opérateur de Koopman. La variante duale est utilisée car le nombre de paramètres à estimer est plus important que le nombre de mesures

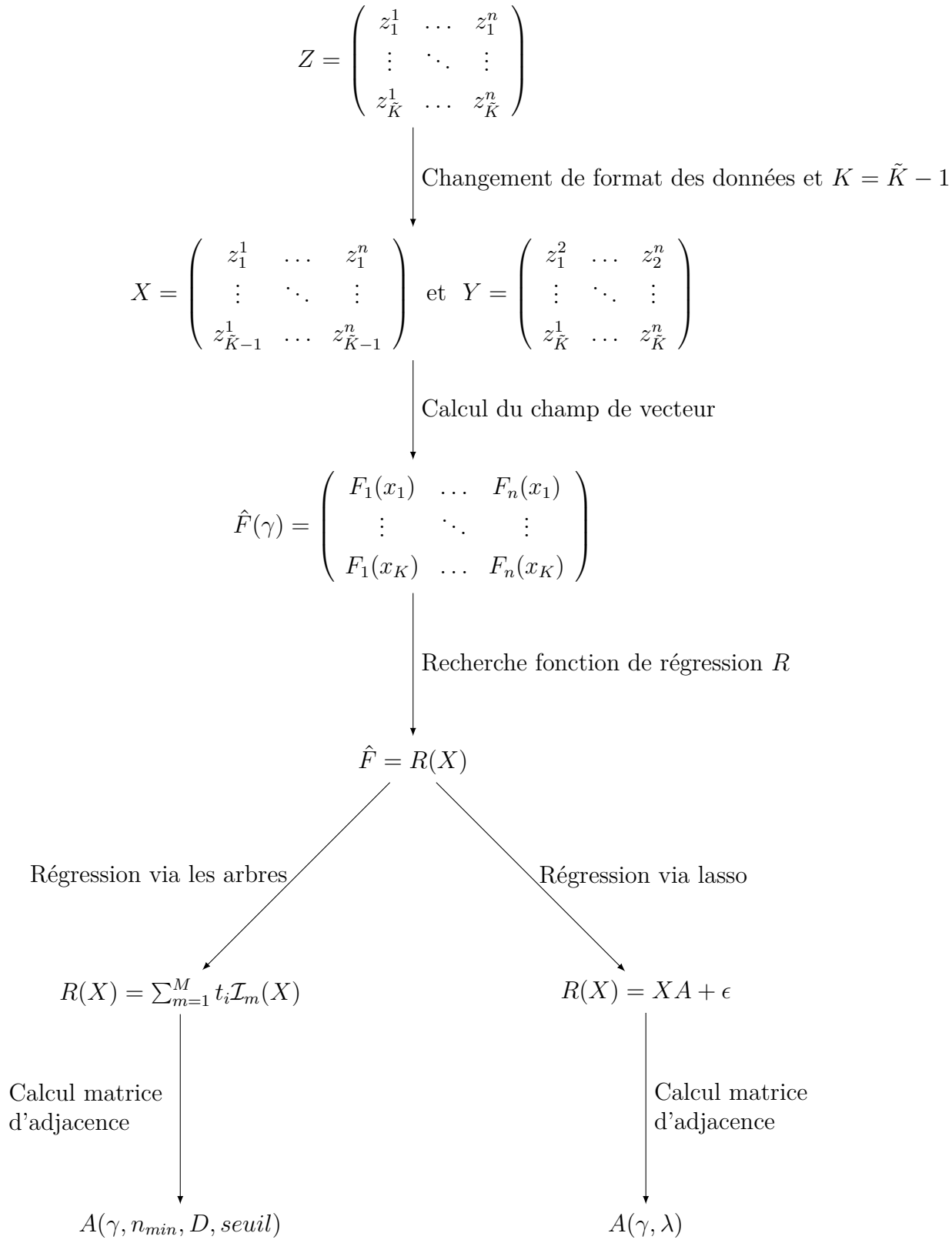


FIGURE 2.3 – Récapitulatif de la méthodologie

temporelles. Celle-ci dépend d'un paramètre γ pour fixer les fonctions de bases.

Ensuite, une régression est réalisée sur le champ de vecteur pour l'expliquer en fonction des données X, c'est-à-dire que la variabilité des x^i est exprimée en fonction des x^i. La régression est effectuée de manière indépendante pour chacune des composantes de $\hat{F}$, il y a donc n problèmes de régression. Pour cela, deux méthodes sont utilisées, la première via les arbres décisionnels est a priori efficace pour tout type d'interactions, mais son efficacité a été montrée pour de grands jeux de données. La matrice d'adjacence A est calculée en considérant l'influence de chaque variable sur la fonction score qui permet de construire l'arbre, celle-ci dépend de 3 paramètres en plus de γ : n_{min}, D et un seuil. La seconde utilise la régression linéaire sous contrainte et permet de sélectionner les variables les plus explicatives du modèle, elle dépend du paramètre λ. La matrice A est construite via les variables sélectionnées lors de la régression.

CHAPITRE 3

ÉVALUATION DE LA MÉTHODE SUR DES DONNÉES SYNTHÉTIQUES

Pour pouvoir évaluer la méthode d'identification de réseau basée sur l'opérateur de Koopman, nous allons utiliser un jeu de données synthétiques. Dans un premier temps, nous décrirons comment ces données ont été obtenues. Nous appliquerons ensuite la méthode d'identification de réseau proposée au chapitre 2 et regarderons l'influence du paramètre γ. Les performances de la méthode lifting seront comparées avec une identification de la dynamique via les différences centrées. Finalement, nous proposerons des pistes d'amélioration.

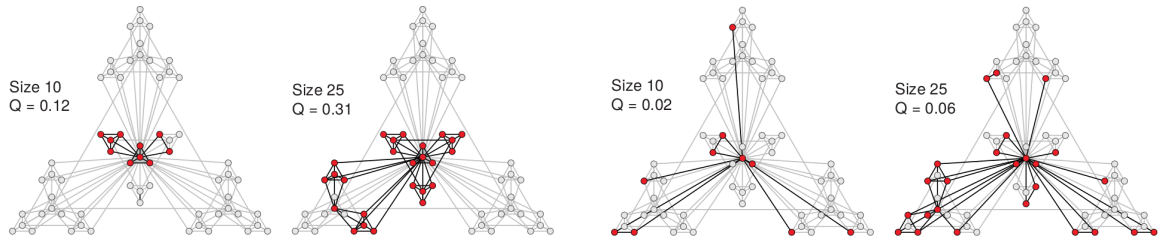
3.1 Contexte

Pour pouvoir évaluer notre méthode, nous utiliserons les données synthétiques du concours *DREAM 4 - In Silico Network Challenge* [3] organisé en 2009. La construction de ces données est décrite dans les articles [19], [20] et [24]. L'objectif de ce concours était d'inférer des réseaux de régulation génétique à partir de données simulées. Ces données correspondent à des séries temporelles du taux d'ARNm dans chaque gène.

Les réseaux utilisés n'ont pas été générés aléatoirement, mais de manière à être les plus réalistes possible, c'est-à-dire proches des structures de réseaux de régulation génétique. Pour cela, les organisateurs du concours ont utilisés des parties de réseaux biologiques déjà identifiés. Ces éléments du réseaux sont sélectionnés de la manière suivante : à partir d'un nœud, d'autres nœuds voisins sont progressivement ajoutés jusqu'à obtenir un réseau de la taille souhaitée. Ces nœuds sont choisis pour maximiser la modularité Q définie par

$$Q = N_1 - N_2$$

où N_1 est le nombre de nœuds dans le sous-réseau et N_2 le nombre de ces nœuds attendus dans un sous-réseau aléatoire. De cette manière, des modules du réseau sont identifiés comme illustré à la figure 3.1. Pour construire le réseau complet, plusieurs modules ainsi construits sont associés.



(a) Identification de modules (b) Identification aléatoire

FIGURE 3.1 – Illustration de l'identification des sous-réseaux.
Source : [19].

Une fois que le réseau a été identifié, il faut générer les données qui permettront de tester les méthodes d'identification sur ce réseau connu. Pour cela, le système est modélisé sous la forme d'équations différentielles exprimant le taux d'ARNm x_i et celui de protéines y_i dans chacun des gènes. Les équations différentielles sont données par

$$\frac{dx_i}{dt} = m_i f_i(y) - \lambda_i^{ARNm} x_i \tag{3.1}$$

$$\frac{dy_i}{dt} = r_i x_i - \lambda_i^{Prot.} y_i \tag{3.2}$$

où λ_i^{ARNm} et λ_i^{Prot} sont les taux de décroissance et la fonction f_i détermine le taux d'activation du gène i. Elle se situe entre 0 (lorsque le gène n'est pas activé) et 1 (lorsque le gène est activé au maximum). Les différentes séries temporelles sont construites via une intégration numérique de ces équations à partir de conditions initiales différentes. Les concentrations en protéines sont utilisées pour construire les données, mais seules les concentrations en ARNm sont mises à disposition pour identifier les réseaux afin de correspondre au problème réel. Bien que les données soient construites de manière à correspondre le plus possible à la réalité, elle restent une représentation simplifiée des réseaux génétiques [24].

Pour être réalistes, les données sont également bruitées. Le bruit interne, c'est-à-dire celui correspondant aux variations du système lui-même, est ajouté. Celui-ci est également calculé via des équations différentielles, mais contrairement aux équations (3.1) et (3.2), celles utilisées pour l'estimation du bruit sont stochastiques. Finalement un bruit externe correspondant aux mesures est ajouté. Celui-ci est une combinaison d'un bruit normal et log-normal qui est un des modèles pour représenter le bruit des puces à ADN.

Le jeu de données *DREAM-4* comprend 10 réseaux à identifier, cinq de taille 10 et cinq de taille 100. Pour les réseaux de taille 10, il y a 5 séries temporelles de 21 points et pour ceux de taille 100, il y en a 10. Les séries temporelles sont construites de la manière suivante : au temps $t = 0s$, une perturbation est appliquée. La mesure du taux d'ARNm se fait toutes les 50 secondes. Lorsque $t = 500s$, la perturbation est retirée, le système revient donc à sa situation initiale. Le taux d'ARNm est toujours mesuré jusqu'à ce que $t = 1000s$. Les données ont ensuite été normalisées.

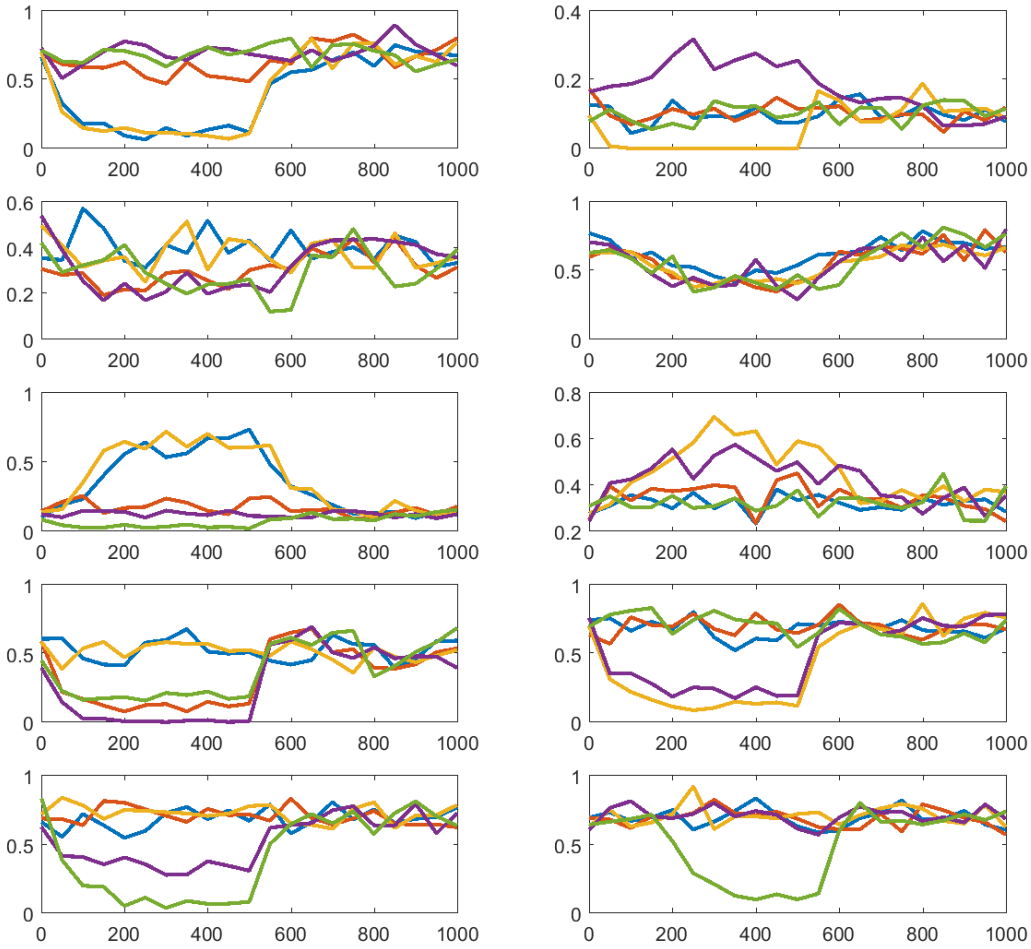


FIGURE 3.2 – Taux d'ARNm dans chacun des gènes pour le premier réseau de taille 10 et pour les 5 séries temporelles.

À titre d'exemple, la figure 3.2 illustre les données pour chaque gène pour le premier réseau de taille 10. Les différentes courbes représentent les différentes séries temporelles. Nous observons que les différentes séries n'ont pas le même comportement. Cela s'explique par le fait que des perturbations différentes sont appliquées et parce-que le bruit a un comportement stochastique. Nous constatons que les données sont effectivement bruitées. Nous envisagerons par la suite de lisser les données initiales pour espérer avoir de meilleurs résultats.

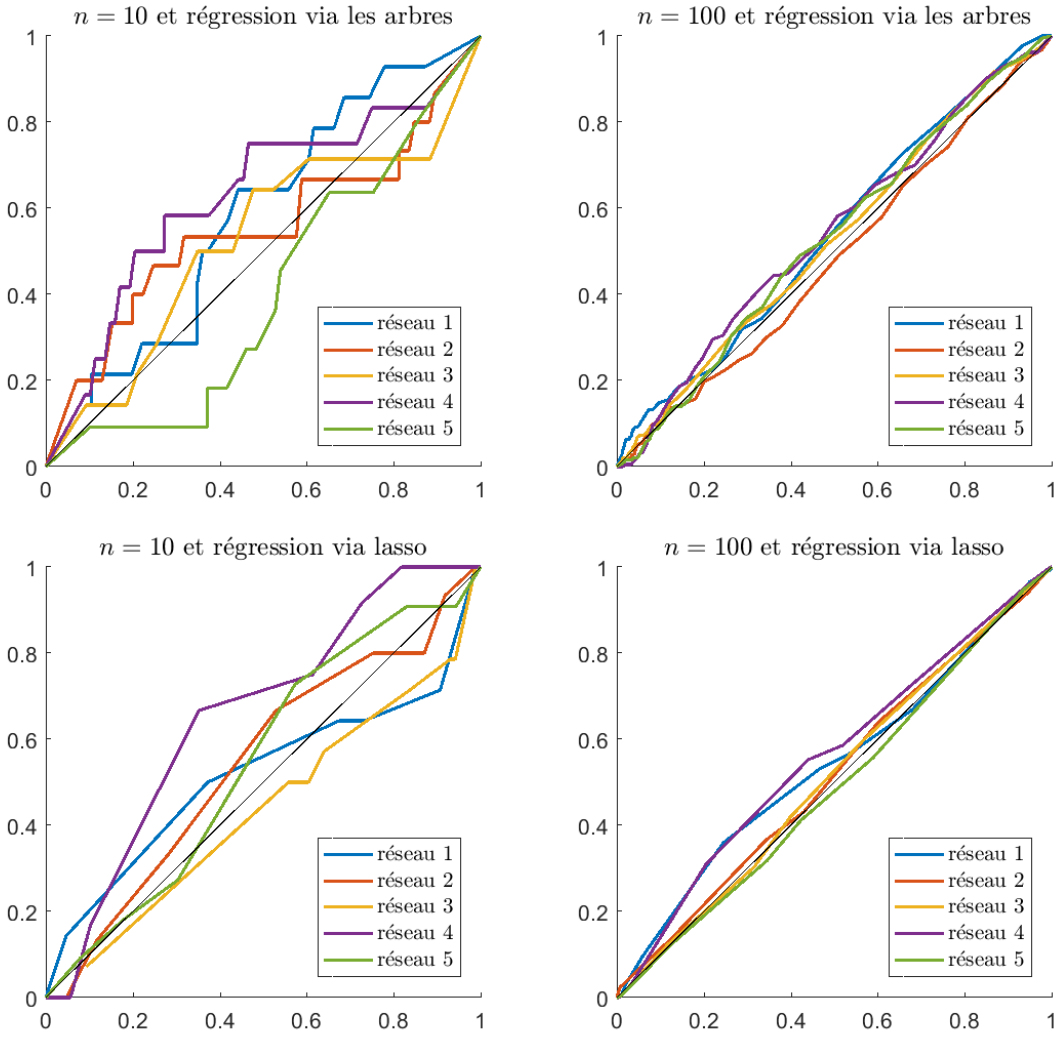
3.2 Évaluation

Dans cette section, nous allons appliquer la méthode d'identification de réseau comme décrite au chapitre 2. Pour commencer, nous analyserons les performances en fonction de la taille du réseau n et du choix de régression. Le jeu de données *DREAM-4* permet de tester deux possibilités : $n = 10$ et $n = 100$. Lors du calcul du champ de vecteurs avec l'opérateur de Koopman, nous utiliserons la variante duale. Il faut donc fixer le paramètre γ pour construire les fonctions g_k données par (2.17), nous discuterons de ce choix. Pour utiliser la régression via les arbres, il faut choisir n_{min} et D comme expliqué à la section 2.2.1. Pour ce jeu de données, nous avons choisi de conserver $n_{min} = 5$ et $D = 100$ comme proposé dans l'article [6]. Par la suite, nous comparerons les résultats obtenus avec une autre configuration de paramètres.

Pour évaluer les performances, nous utiliserons la courbe ROC, il ne faut donc ni fixer de seuil pour la méthode des arbres, ni λ pour la méthode lasso, car ceux-ci varient. La figure 3.3 reprend les courbes ROC pour les différents réseaux en fonction de la taille et des deux méthodes de régression. Le paramètre γ utilisé vaut 0.01 et nous discuterons de ce choix dans la section suivante.

Commençons par analyser les résultats pour les réseaux de taille 10. Nous observons que les performances de la méthode varient fortement en fonction du réseau, les valeurs AUROC se situent entre 0.40 et 0.66. Le cinquième réseau a une valeur AUROC en deçà de 0.5 pour la régression via les arbres. Cela signifie que son identification est moins efficace que ce que nous pourrions attendre d'une identification aléatoire. C'est également la cas du réseau 3 avec la régression via lasso. À l'opposé le réseau 4 est le mieux identifié, l'aire sous la courbe ROC est de 0.63 pour la régression via les arbres de 0.66 pour celle réalisée avec lasso. L'identification de réseaux dépend très fortement de leur structure. Les méthodes d'identification sont plus efficaces pour certains réseaux que pour d'autres [24]. Cela signifie qu'un des objectifs de l'identification est également d'avoir une faible variance entre les résultats obtenus avec différents réseaux. Dans notre cas de figure, cet objectif n'est pas atteint.

En moyenne, les deux méthodes de régression ont des résultats similaires. En effet, la moyenne des valeurs AUROC est 0.54 pour les deux régressions. Notons également que les résultats pour la régression via les arbres peuvent varier d'une exécution à une autre puisqu'elle contient une part d'aléatoire. Pour la régression via le lasso, l'unicité est garantie puisque le nombre de variable de régression n est inférieur au nombre d'observations K.



	n=10					n=100				
Abres	0.58	0.55	0.52	0.63	0.4	0.54	0.48	0.53	0.54	0.53
Lasso	0.52	0.55	0.43	0.66	0.55	0.54	0.52	0.51	0.56	0.49

FIGURE 3.3 – Courbes ROC et valeurs AUROC obtenues en identifiant le champ de vecteurs via la méthode lifting duale pour les données *DREAM-4*.

Lorsque $n = 100$, nous observons moins de variabilité des performances en fonction des réseaux. Pour $n = 10$ les valeurs AUROC étaient compris entre 0.40 et 0.66 tandis que lorsque $n = 100$, elles sont comprises entre 0.49 et 0.56. Cela s'explique par le fait que le nombre de liens possibles est plus grand et donc les différents taux calculés (TVP et TFP) sont des résultats plus moyens. Les valeurs AUROC se situent proches des résultats d'une identification aléatoire bien que la moyenne soit légèrement supérieure à 0.5 : elle est de 0.52 pour les deux régressions.

Au regard des analyses réalisées pour $n = 10$ et $n = 100$, il n'est pas possible de déterminer si une des deux régressions proposées est plus efficace. Les deux stratégies mènent à des résultats similaires.

À ce stade, nous ne pouvons pas conclure que la méthode d'identification de réseau basée sur l'opérateur de Koopman est efficace. En effet, nous observons que lorsque $n = 10$, les résultats sont très variables et pas toujours mieux que de l'aléatoire. Lorsque $n = 100$, les résultats sont fort proches de l'aléatoire bien que légèrement supérieurs. Pour tenter soit d'améliorer les performances de la méthode, soit de confirmer ces résultats, nous allons investiguer plusieurs pistes. Premièrement, nous essayerons de trouver un meilleur choix pour le paramètre γ. Nous comparerons ensuite les résultats avec ceux obtenus en calculant le champ de vecteurs via les différences centrées. Finalement, nous proposerons des pistes d'amélioration pour la méthode d'identification.

3.2.1 Choix du paramètre γ

Les résultats de la figure 3.3 ont été obtenus en fixant $\gamma = 0.01$. Ce choix a été motivé par les résultats de l'article [21] et par des tests préliminaires à l'évaluation de la méthode. Néanmoins, au vu des résultats obtenus, nous sommes en mesure de nous demander si ce choix de γ était le bon. Pour rappel, γ intervient dans la définition des fonctions Gaussiennes g_k (2.17) qui permettent d'identifier le champ de vecteurs, il a donc un rôle déterminant. Au plus γ est petit, au plus les fonctions de bases se rapprocheront de constantes.

Pour choisir ce paramètre, il n'existe pas de critère, c'est pourquoi nous allons rechercher la meilleure valeur de manière heuristique. L'identification de réseaux a été réalisée pour différentes valeurs de γ et les valeurs AUROC obtenues en fonction de γ, n et du réseau en particulier se trouvent à la figure 3.4. Les meilleurs performances et les moins bonnes sont respectivement indiquées en vert et en rouge pour chacun des réseaux et pour les deux méthodes de régression. Les résultats moyens en fonction de γ sont également indiqués.

Le premier constat que nous pouvons faire c'est que les résultats pour un même paramètre γ sont très variables en fonctions du réseau. Considérons par exemple le cas où $\gamma = 0.001$ avec la régression via les arbres. Le premier réseau donne une aire sous la courbe ROC de 0.6 tandis que pour le quatrième réseau, la valeur AUROC n'est que de 0.3.

$n = 10$														
	Régression via les arbres							Régression via lasso						
γ	1	0.5	0.1	0.05	0.01	0.005	0.001	1	0.5	0.1	0.05	0.01	0.005	0.001
1	0.46	0.51	0.45	0.54	0.48	0.48	0.6	0.46	0.46	0.64	0.55	0.52	0.63	0.48
2	0.49	0.34	0.32	0.57	0.63	0.63	0.6	0.56	0.56	0.3	0.53	0.55	0.57	0.53
3	0.63	0.57	0.59	0.57	0.45	0.43	0.49	0.48	0.4	0.47	0.52	0.43	0.58	0.36
4	0.49	0.49	0.42	0.61	0.64	0.44	0.3	0.48	0.52	0.49	0.56	0.66	0.54	0.46
5	0.45	0.43	0.48	0.35	0.42	0.58	0.4	0.5	0.37	0.5	0.56	0.55	0.64	0.66

$n = 100$														
	Régression via les arbres							Régression via lasso						
γ	1	0.5	0.1	0.05	0.01	0.005	0.001	1	0.5	0.1	0.05	0.01	0.005	0.001
1	0.6	0.55	0.55	0.59	0.57	0.53	0.6	0.5	0.5	0.5	0.54	0.54	0.52	0.56
2	0.44	0.47	0.52	0.48	0.52	0.44	0.37	0.5	0.47	0.5	0.51	0.52	0.51	0.48
3	0.53	0.52	0.52	0.56	0.52	0.56	0.51	0.47	0.51	0.49	0.51	0.52	0.51	0.5
4	0.51	0.5	0.49	0.54	0.54	0.51	0.53	0.49	0.51	0.55	0.55	0.56	0.56	0.57
5	0.49	0.51	0.43	0.54	0.48	0.5	0.45	0.52	0.5	0.49	0.49	0.49	0.50	0.51

Résultats moyens en fonction de γ														
γ	1	0.5	0.1	0.05	0.01	0.005	0.001	1	0.5	0.1	0.05	0.01	0.005	0.001
	0.51	0.49	0.48	0.55	0.53	0.51	0.49	0.50	0.48	0.49	0.53	0.53	0.56	0.51

FIGURE 3.4 – Valeurs AUROC en fonction du paramètre γ pour les données *DREAM-4*.

Ces résultats nous confirment également que l'identification de réseau est sensible au choix du paramètre γ. Par exemple, pour le cinquième réseau de taille 10 et pour la régression via lasso, l'aire sous la courbe ROC est de 0.66 avec $\gamma = 0.001$, tandis qu'elle est de 0.37 lorsque $\gamma = 0.5$. Cet exemple nous montre qu'en fonction du paramètre γ, l'identification est considérablement différente et peut mener à des résultats mauvais (en deçà de ceux espérés avec une identification aléatoire) ou alors satisfaisants.

Bien que γ joue un rôle sur les performances de la méthode, il n'y a pourtant pas de valeur de γ qui semble clairement meilleure que les autres. Nous pouvons tout de même rechercher quel choix serait optimal.

Commençons par le cas de la régression via les arbres. Nous observons que les valeurs de γ de 0.05, 0.01 et 0.005 sont celles qui permettent le plus d'avoir les meilleures performances. Les choix de $\gamma = 0.1$ ou $\gamma = 0.001$ mènent a des valeurs bien en-dessous de 0.5 pour les valeurs AUROC, ils sont donc à éviter. Pour la régression via la méthode lasso, les meilleurs choix de paramètres sont $\gamma = 0.01$ et $\gamma = 0.005$. Notons que les résultats obtenus avec $\gamma = 0.05$ restent également proches de ceux-ci. Par contre, pour cette régression, nous observons que lorsque γ est supérieur ou égal à 0.01, les valeurs AUROC sont plus petites. Nous exclurons donc ces choix.

Au final, il apparaît que les valeurs de γ comprises entre 0.05 et 0.005 sont celles qui permettent, en moyenne, la meilleure identification. Cela n'exclut pas que pour certains réseaux, d'autres valeurs sont plus performantes. Lors des prochains tests nous conserverons $\gamma = 0.01$.

Regardons maintenant les résultats moyens : les valeurs AUROC moyennes se situent entre 0.48 et 0.56. Cela nous montre que la méthode d'identification basée sur l'opérateur de Koopman est au mieux légèrement plus efficace d'une identification aléatoire. Néanmoins les performances restent peu satisfaisantes.

3.3 Comparaison avec l'approximation du champ de vecteurs calculé via les différences finies

Désormais, nous allons, à nouveau, identifier les mêmes réseaux mais, cette fois-ci, pour calculer le champ de vecteur, nous utiliserons les différences finies. Nous comparerons les résultats obtenus avec ceux calculés via la méthode lifting.

Considérons les données temporelles

$$z_k = z_k^1, ..., z_k^n \qquad \forall k = 1, ..., \tilde{K},$$

toutes mesurées à un intervalle de temps T_s. Alors, le champ de vecteur calculé via les différences finies centrées peut être approximé par

$$\hat{F}(z_k) = \frac{z_{k+1} - x_{z-1}}{2T_s} \qquad \forall k = 2, ..., \tilde{K} - 1.$$

En pratique, comme expliqué à la section 3.1, les données sont sous la forme de plusieurs séries temporelles et il faut donc calculer le champ de vecteur séparément pour chacune.

Pour comparer les performances avec la méthode lifting, nous utiliserons également les courbes ROC. Les résultats obtenus se trouvent à la figure 3.5.

Nous observons directement que l'identification réalisée avec l'approximation du champ de vecteurs via les différences centrées est plus efficace que lorsque celui-ci était calculé via la méthode lifting. Nous trouvons des valeurs AUROC entre 0.53 et 0.71 tandis que précédemment, elles se situaient entre 0.40 et 0.66. Désormais, toutes les identifications ont une valeur AUROC supérieure à 0.5, ce qui est souhaité. De plus les meilleurs valeurs AUROC sont de l'ordre de 0.7.

Analysons le cas où $n = 10$, nous observons que les résultats entre les différents réseaux sont plus variables pour la régression avec les arbres (AUROC entre 0.53 et 0.69) qu'avec lasso (AUROC entre 0.56 et 0.71). En moyenne, les résultats via les deux régressions sont très proches : l'AUROC moyenne pour lasso est de 0.61 et de 0.62 pour les arbres. Nous ne pouvons pas dire, dans ce cas, si une des deux régressions est plus efficace.

Concernant les réseaux de dimension $n = 100$, les deux régressions mènent également à des résultats assez proches (AUROC de 0.65 en moyenne pour les arbres et de 0.62 pour lasso). Par contre les réseaux de taille 100 sont systématiquement mieux identifiés via l'algorithme basé sur les arbres décisionnels. Cette régression semble donc préférable dans ce cas.

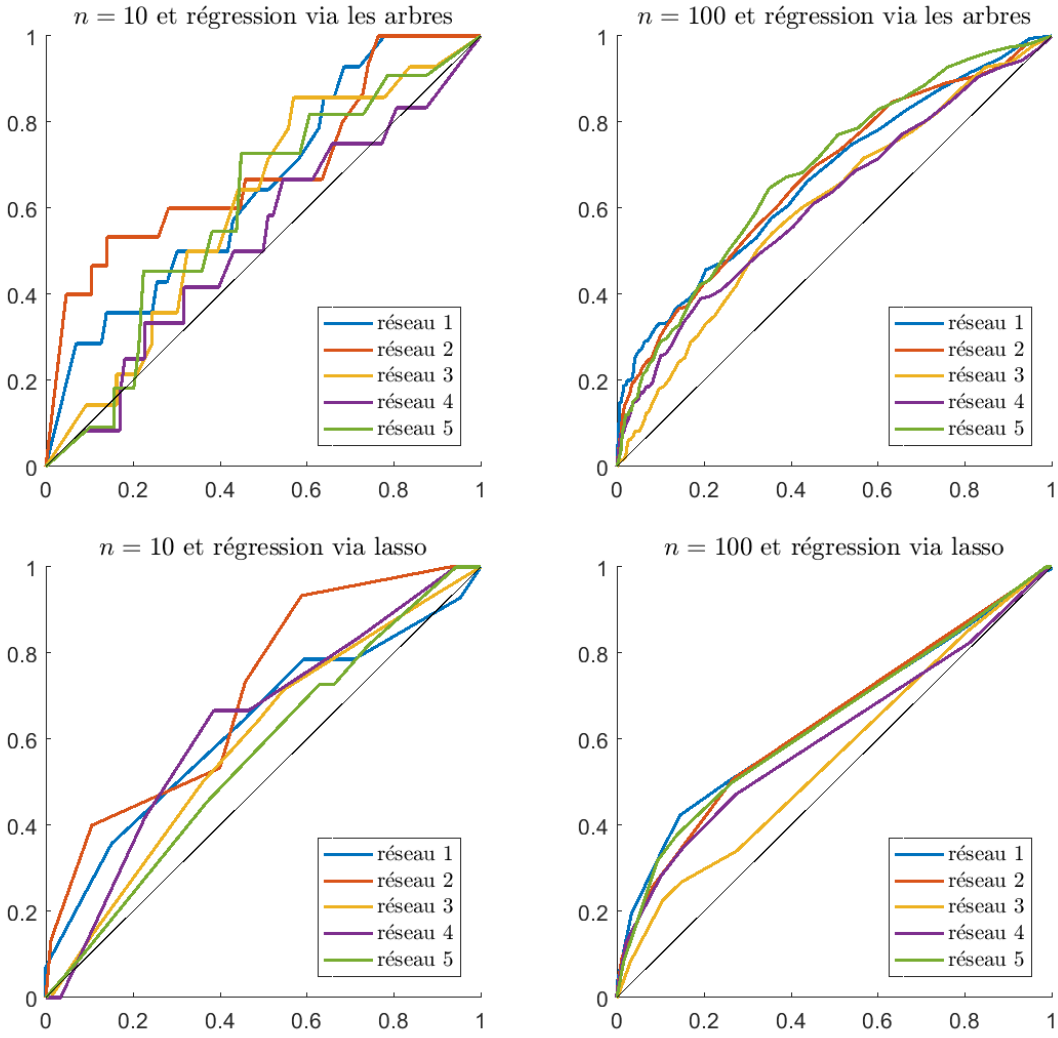
En conclusion, il apparaît clairement que la méthode d'identification est plus efficace lorsque nous identifions le champ de vecteurs via les différences centrées. Nous observons des valeurs AUROC de l'ordre de 0.6 en moyenne et dans certains cas, elles atteignent l'ordre de 0.7. Cela est déjà un bon apport au vu de la complexité de la tâche et des données. Nous placerons ces résultats dans le contexte du concours à la section 3.5.

3.4 Pistes de réflexion

Nous venons de voir que la méthode lifting avec la variante duale ne permettait pas une identification de réseau performante. Tandis qu'en utilisant les différences centrées, nous parvenons à obtenir des valeurs AUROC significativement supérieures à 0.5. Dans cette section, nous tenterons d'apporter des modifications à la méthode d'identification de réseau pour augmenter ses performances.

3.4.1 Utilisation de la variante principale lorsque $n = 10$

Dans le chapitre 2, nous avons présenté deux variantes pour la méthode lifting : la version principale et la version duale. Toutes les deux divergent par le choix des fonctions de bases. Dans un premier temps, nous nous sommes concentrés sur la méthode duale. Celle-ci est appliquée lorsque $K \leq N$ où K est le nombre de paires de données



	n=10					n=100				
Arbres	0.65	0.69	0.6	0.53	0.6	0.67	0.67	0.61	0.62	0.69
Lasso	0.63	0.71	0.59	0.63	0.56	0.65	0.64	0.56	0.6	0.64

FIGURE 3.5 – Courbes ROC et valeurs AUROC obtenues en identifiant le champ de vecteurs via les différences centrées pour les données *DREAM-4*.

vérifiant (2.4) et N le nombre de paramètres à inférer. Lorsque nous utilisons la méthode duale, nous fixons $K = N$ et la dynamique du système est approchée avec les fonctions de bases Gaussiennes g_k définie par (2.17).

Le nombre de paramètres à estimer N dépend, en règle générale, de la dimension du système n. Lorsque nous travaillons avec des systèmes pour lesquels n est petit, il est possible d'utiliser la variante principale. Celle-ci nécessite que $K \geq N$. Dans ce cas, nous choisissons comme fonctions de bases les monômes p_k dont le degré est inférieur ou égal à m. Le nombre de paramètres à estimer N dépend également de m, c'est pourquoi il faut également limiter m pour garantir la condition $K \geq N$.

Pour les jeux de données de taille $n = 10$, nous pouvons utiliser la méthode principale. En effet, chaque série temporelle contient 21 points, et permet de construire 20 paires de données vérifiant (2.4). En utilisant les 5 séries temporelles, nous obtenons $K = 100$. Si nous choisissons les monômes de degré maximal 2, alors le nombre de paramètre à inférer est donné par (2.10) et vaut

$$N = \frac{n + m!}{n!m!} = \frac{12!}{10!2!} = 66.$$

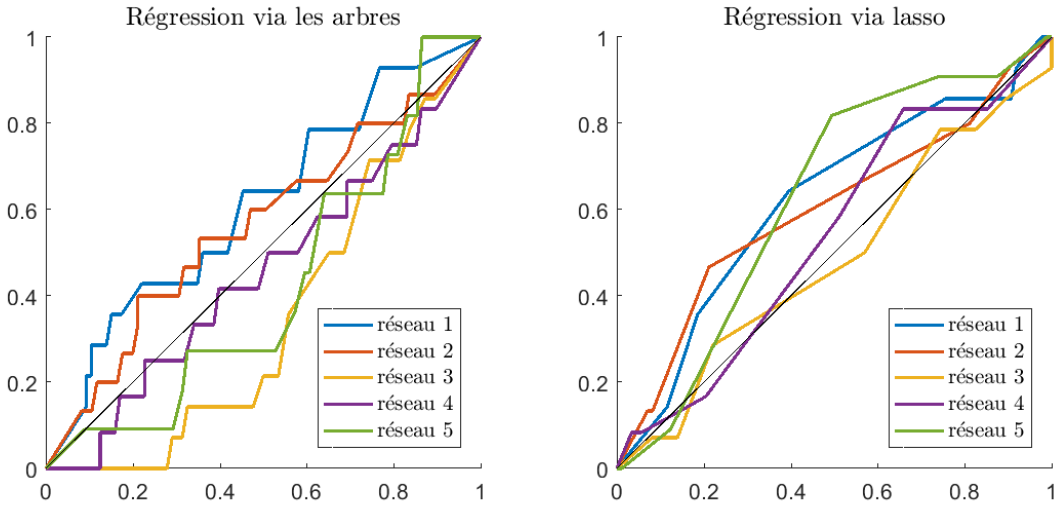
Nous vérifions donc bien que $K \geq N$. Si nous considérons $m > 2$ alors la condition $K \geq N$ n'est plus vérifiée, c'est pourquoi nous n'utiliserons pas de plus grandes valeurs pour m. Si nous voulions utiliser la méthode principale lorsque $n = 100$, nous serions limités par le nombre d'observations et m ne pourrait excéder 1. Autrement dit, la méthode lifting principale reviendrait à une régression linéaire classique.

L'identification a donc été réalisée via la variante principale pour les réseaux de taille 10. Celle-ci ne nécessite plus de choisir le paramètre γ. Les courbes ROC obtenues se trouvent à la figure 3.6.

Nous observons directement que la régression via lasso est plus performante que celle via les arbres. En moyenne l'AUROC est de 0.5 pour les arbres et de 0.58 pour lasso. Dans le cas de la régression avec les arbres, les résultats pour le réseau 3 sont particulièrement mauvais puisque l'AUROC est seulement de 0.35. Autrement dit, pour avoir une bonne identification, il faudrait faire l'opposé de ce que la méthode suggère. De plus, seule la méthode lasso se montre, en moyenne, plus efficace que de l'aléatoire. Notons que la méthode lasso réalise une régression linéaire or la méthode lifting principale approxime le champ de vecteur selon des monômes. Il est donc plus aisé d'effectuer ce type de régression.

Nous pouvons conclure que, lorsque le réseau et de dimension 10 et que le champ de vecteurs est calculé via la méthode lifting principale, il est préférable de réaliser la régression via lasso.

Ces résultats peuvent être comparés avec ceux obtenus via la méthode lifting duale et via les différences centrées se trouvant respectivement aux figures 3.3 et 3.5. Les valeurs moyennes de AUROC sont reprises à la table 3.1. Pour chaque stratégie de



	n=10				
Arbres	0.61	0.56	0.35	0.45	0.42
Lasso	0.62	0.6	0.49	0.54	0.63

FIGURE 3.6 – Courbes ROC et valeurs AUROC obtenues en identifiant le champ de vecteurs via la méthode lifting principale pour les données *DREAM-4*.

	lifting dual	lifting principal	différences centrées
Arbres	0.54	0.48	0.61
Lasso	0.54	0.58	0.62

TABLE 3.1 – Comparaison des valeurs AUROC moyennes pour la variante principale de la méthode lifting

calcul du champ de vecteurs, la régression via lasso est plus efficace ou égale à celle via les arbres. La différence est surtout marquée pour le cas de la méthode lifting avec la variante principale. Dans les autres cas, les résultats sont très proches.

Concernant le choix de la méthode pour identifier la dynamique, la variante principale de lifting permet d'obtenir de meilleur résultats que la variante duale lorsque la régression est réalisée via lasso. Néanmoins les performances des différences centrées restent au-dessus des performances de la méthode lifting, que se soit pour la variante principale ou duale. Autrement dit, l'utilisation de la variante principale ne permet pas à la méthode lifting de dépasser les résultats obtenus avec le calcul du champ de vecteurs via les différences centrées.

3.4.2 Modification du calcul du logarithme matriciel au sein de l'identification du champ de vecteurs

Lors de l'identification du champ de vecteurs via la méthode lifting, nous devons utiliser un logarithme matriciel pour calculer la matrice L, que cela soit dans la version principale (2.16) ou duale (2.20). Ce calcul ne peut se faire que sous certaines conditions que nous allons analyser. Commençons par rappeler la condition pour qu'un logarithme matriciel soit réel.

Soit l'équation

$$e^X = A,$$

où X désigne le logarithme matriciel de A. L'équation admet une solution unique si et seulement si la condition suivante est vérifiée :

$$\xi_i \notin \mathbb{R}^- \quad \forall \xi_i \text{ valeur propre de } A. \tag{3.3}$$

Ce logarithme est alors appelé logarithme principal et il vérifie la condition suivante

$$-\pi < Im(\xi_i) < \pi$$

où ξ_i désigne les valeurs propres de A [11]. Lorsque A est réel, son logarithme principal X est réel aussi. Au contraire, si la condition (3.3) n'est pas respectée, alors X est complexe.

Lors du calcul de L, nous calculons le logarithme matriciel de $P_y P_x^\dagger$ (ou $P_x^\dagger P_y$ pour les tests avec la méthode principale), nous noterons cette matrice A. Nous désirons que A soit réel mais il n'y a pas de moyen de garantir que A vérifie (3.3). D'ailleurs en pratique, pour la variante duale, nous observons qu'elle contient un nombre de valeurs propres réelles strictement négatives. Ce nombre est assez faible (entre 1 et 5) par rapport à la taille de la matrice A (100×100). Les nombres de valeurs propres ne vérifiant pas (3.3) pour chaque réseau sont repris à la table 3.2. Dans ce cas, L est complexe et la méthode d'identification néglige sa partie imaginaire. Cela implique donc une source d'erreur. Malheureusement, celle-ci provient des données et il n'est donc pas possible de l'éviter.

	réseau 1	réseau 2	réseau 3	réseau 4	réseau 5
$n = 10$	4	2	4	4	5
$n = 100$	5	1	3	2	3

TABLE 3.2 – Nombre de valeurs propres réelles strictement négatives de A pour les différents réseaux

Une piste d'amélioration serait d'artificiellement modifier la matrice L pour "supprimer" les valeurs propres réelles strictement négatives. Nous pouvons décomposer la matrice A en éléments propres

$$A = Q\Lambda Q^{-1}$$

avec Λ la matrice diagonale dont les éléments sont les valeurs propres et Q la matrice contenant les vecteurs propres associés.

Nous pouvons alors modifier la matrice Λ en Λ' dont les éléments diagonaux sont définis par

$$\lambda'_{ii} = \begin{cases} \xi_i \text{ si } \xi_i \notin \mathbb{R}^- \\ \epsilon \text{ sinon.} \end{cases}$$

où ϵ est une valeur arbitraire strictement positive et proche de 0. La matrice A est alors remplacée par

$$A' = Q\Lambda' Q^{-1}.$$

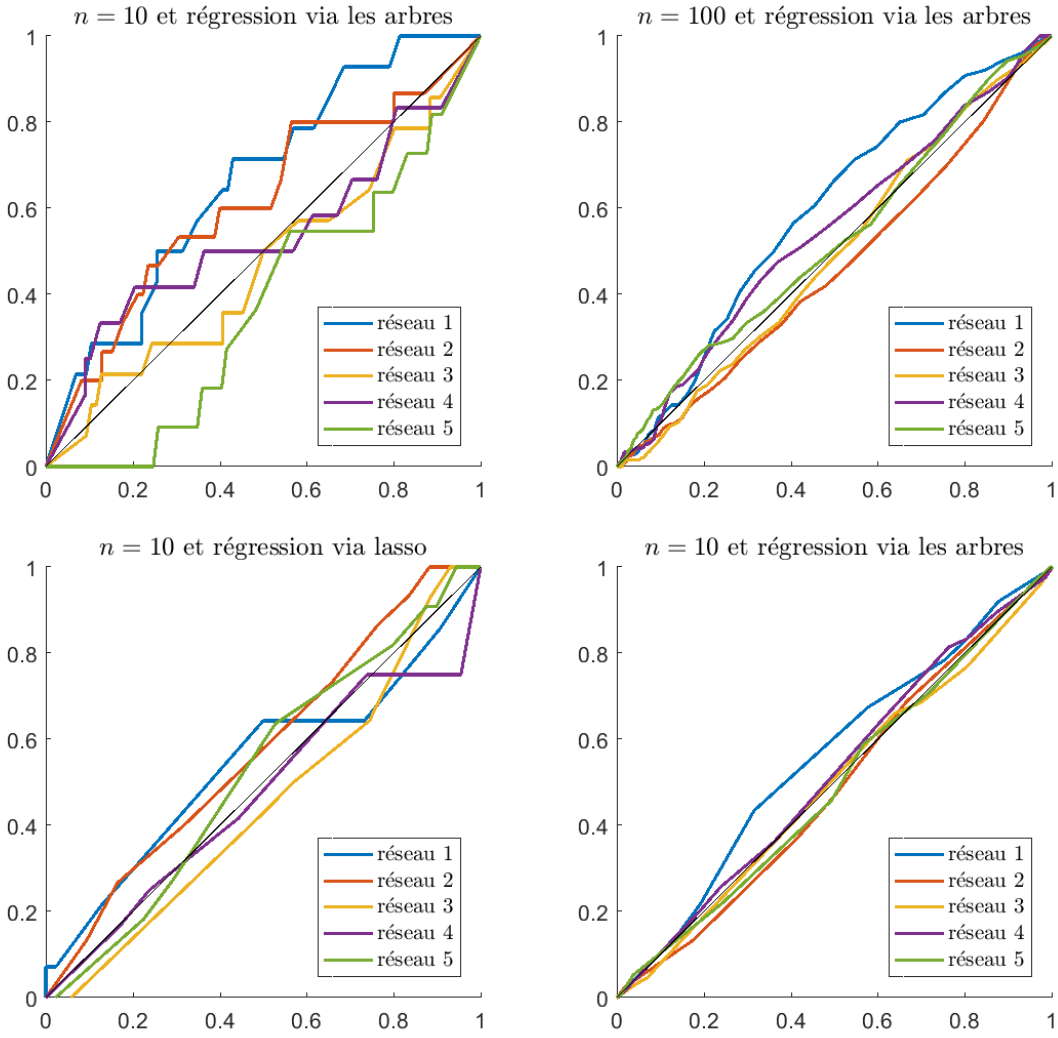
Cette suppression des valeurs propres ne vérifiant pas (3.3) modifie le calcul de L et donc du champ de vecteurs F. Néanmoins, dans ce cas, L est bien réel et en contrepartie, nous ne négligeons pas la partie imaginaire.

L'identification de réseau a été réalisée en modifiant la matrice A comme présenté et les résultats se trouvent à la figure 3.7.

Pour les réseaux de taille 10, les valeurs AUROC sont en moyenne de 0.53 pour la régression via les abres et de 0.51 pour lasso. Bien que les réseaux 1 et 2 aient des valeurs AUROC supérieures à 0.6, l'efficacité de la méthode n'est pas garantie puisque nous observons, par exemple une valeur AUROC de 0.38 pour le cinquième réseau. Nous observons à nouveau une grande variabilité pour les réseau de taille 10 identifiés avec la régression basée sur les arbres.

Lorsque $n = 100$, les résultats sont du même ordre que pour $n = 10$ (0.52 pour la régression via les arbres 0.51 pour lasso). Les résultats de l'identification obtenue en modifiant le calcul matriciel de L sont donc en moyennes similaires à ceux que nous obtiendrions avec une identification aléatoire.

L'objectif de cette modification était d'améliorer les performances de la méthode lifting duale, nous allons donc comparer ces résultats avec ceux de la figure 3.3. À titre de comparaison, les résultats moyens sont repris à la table 3.3. Les résultats des deux



	n=10					n=100				
Arbres	0.66	0.61	0.47	0.53	0.38	0.59	0.46	0.49	0.55	0.52
Lasso	0.54	0.58	0.45	0.47	0.52	0.56	0.49	0.49	0.52	0.49

FIGURE 3.7 – Courbes ROC et valeurs AUROC obtenues en identifiant le champ de vecteur via la méthode lifting modifiée pour les données *DREAM-4*.

	$n = 10$		$n = 100$	
	Sans modif.	Avec modif.	Sans modif.	Avec modif.
Arbres	0.54	0.53	0.52	0.52
Lasso	0.54	0.51	0.52	0.51

TABLE 3.3 – Comparaison des valeurs AUROC moyennes pour la méthode lifting avec et sans modification

identifications sont similaires et toujours proches de ce que nous obtiendrions avec une identification aléatoire. Les conditions d'analyse ne permettent pas d'établir si cette variante est plus efficace puisque dans les deux cas les résultats ne sont pas concluants.

3.4.3 Lissage des données

Dans la section 3.1, nous avons expliqué que les données étaient bruitées. Il y a d'une part la simulation du bruit interne qui est stochastique et le bruit externe. Or, l'efficacité de la méthode lifting n'est garantie que lorsque le bruit tend vers 0. Une autre piste d'amélioration de la méthode serait alors de lisser les données avant de réaliser le calcul du champ de vecteurs.

Pour lisser les données, nous utiliserons la méthode des moyenne mobiles [10]. Celle-ci consiste à remplacer la série temporelle initiale par une nouvelle dont les différents points sont calculés via une moyenne des points adjacents. Les données z_k sont donc remplacées par

$$\tilde{z}_k = \frac{1}{2p+1} \sum_{i=-p}^{p} z_j \qquad \forall k = p+1, ..., \tilde{K} - p$$

où pour rappel $\tilde{K}$ est le nombre de données temporelles et p un paramètre. Les p premiers et derniers éléments sont également calculés via des moyennes, mais le nombre d'éléments utilisés pour calculer la moyenne est limité aux dimensions de la série temporelle. Pour les points centraux de la série temporelle, la moyenne est réalisée sur $2p + 1$ éléments. Lorsque p augmente, la courbe est de plus en plus lisse, ce qui permet d'atténuer le bruit. Néanmoins, au plus le lissage est important, au plus les données seront proches d'une constante et nous risquons alors de perdre de l'information.

Dans un premier temps, le paramètre p est fixé à 2. Les nouvelles données obtenues sont illustrées à la figure 3.8 pour le premier réseau. Nous pouvons les comparer aux données initiales représentées à la figure 3.2. Les irrégularités dues au bruit sont amoindries. Néanmoins, ce lissage déforme également les réactions faces aux perturbations.

Pour tester l'effet de ce lissage, nous allons à nouveau réaliser l'identification des réseaux. Nous recherchons la dynamique via la méthode lifting duale et via les différences

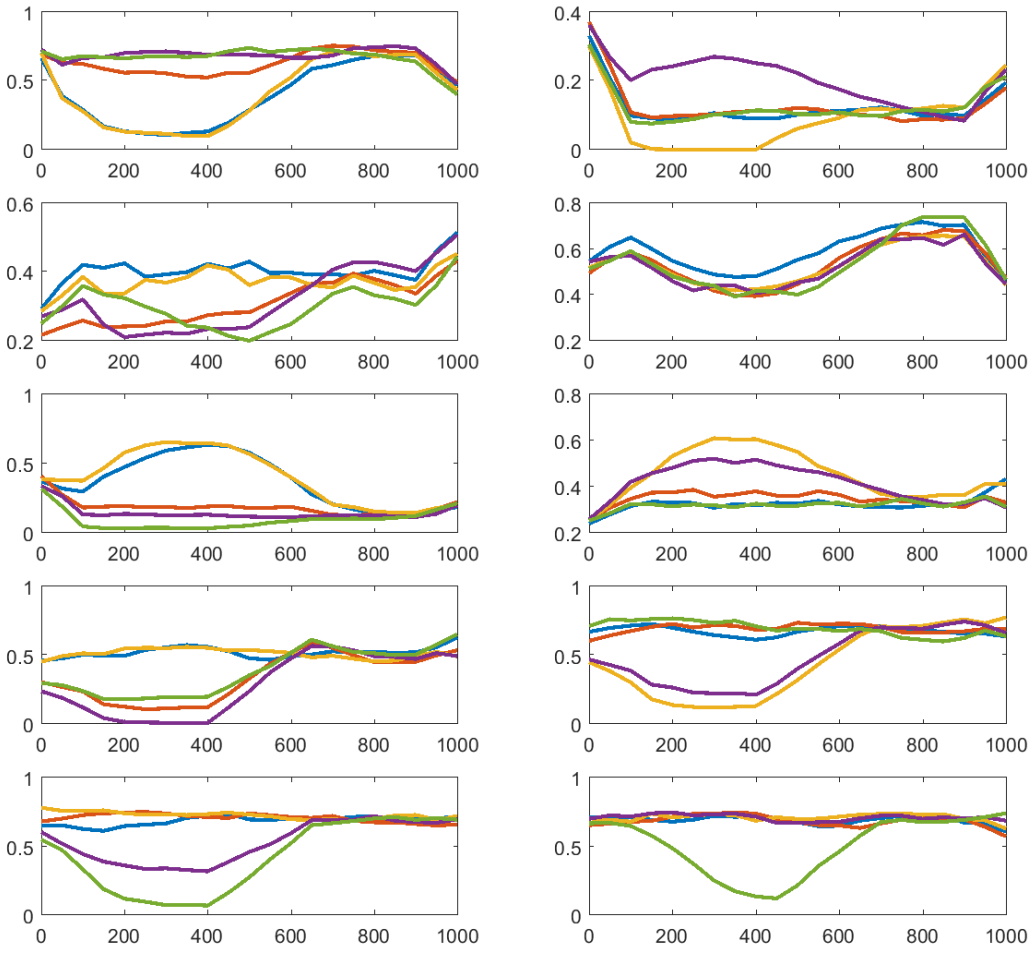


FIGURE 3.8 – Taux d'ARNm dans chacun des gènes pour le premier réseau de taille 10 après lissage des données.

	$n = 10$		$n = 100$	
	Sans liss.	Avec liss.	Sans liss.	Avec liss.
Arbres	0.54	0.49	0.52	0.52
Lasso	0.54	0.57	0.52	0.55

TABLE 3.4 – Comparaison des valeurs AUROC moyennes pour la méthode lifting avec et sans lissage des données.

centrées et comparerons les résultats obtenus.

Commençons par analyser le cas où le champ de vecteurs est estimé grâce à la méthode lifting duale (figure 3.9). Lorsque $n = 10$, la régression via les arbres donne, une nouvelles fois, des résultats fort variables en fonction du réseau. Le réseau 3 a une valeur AUROC de 0.61, tandis qu'elle n'est que de 0.35 pour le réseau 5. Les valeurs AUROC sont plus importantes en utilisant le lasso plutôt qu'avec les arbres (AUROC moyen de 0.57 contre 0.49). De plus, seul le lasso permet d'obtenir des résultats moyens plus performants qu'une méthode aléatoire.

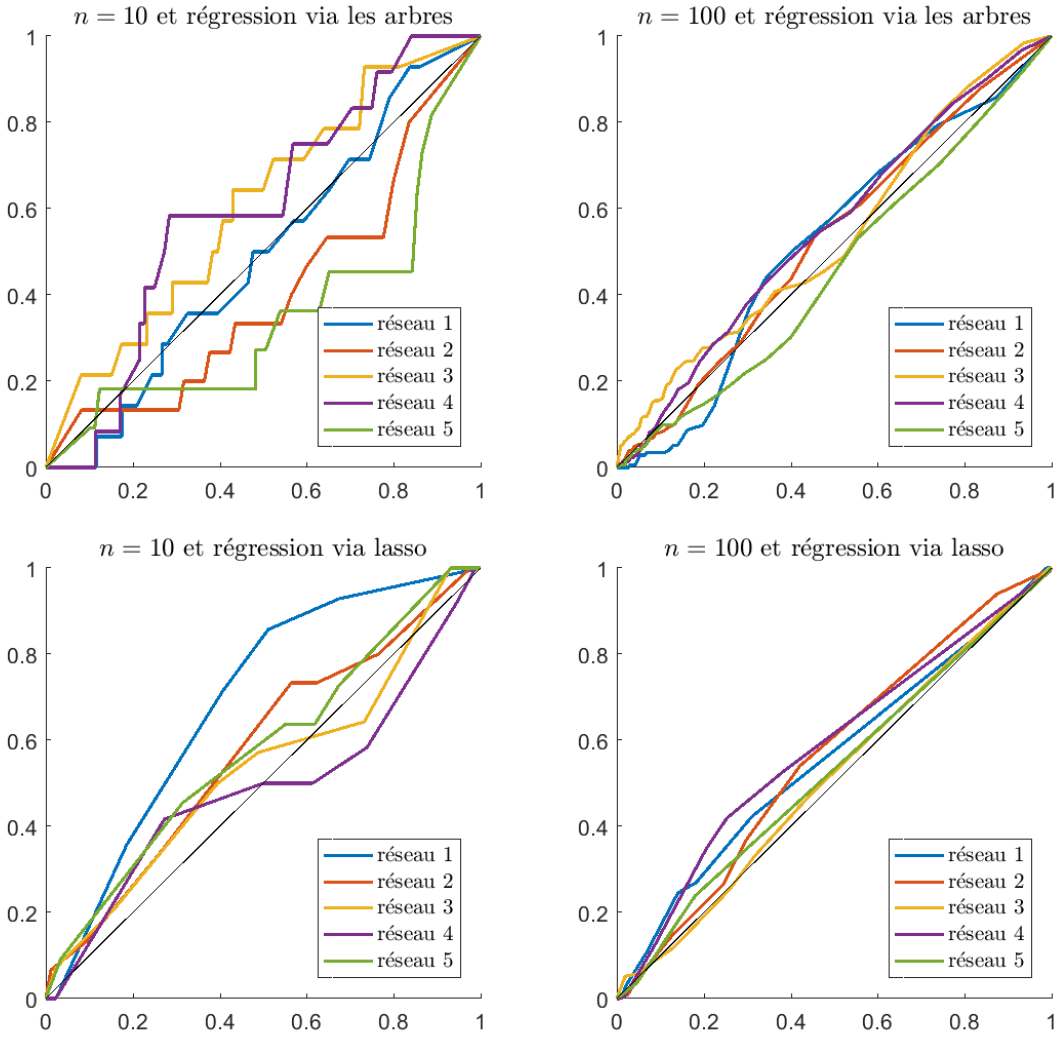
Pour $n = 100$, les valeurs AUROC se situent entre 0.46 et 0.56. Elles sont en moyenne de 0.52 pour la méthode de régression basée sur les arbres et 0.55 pour lasso. Nous observons à nouveau que la lasso est légèrement plus efficace.

L'objectif est d'avoir des résultats meilleurs que pour la même méthode mais sans le lissage des données (voir figure 3.3). Nous allons donc comparer ces deux identifications. Pour cela, les valeurs AUROC moyennes sont reprises à la table 3.4. Nous observons pour la régression via le lasso, que le lissage des données permet d'améliorer légèrement les performances. Néanmoins, globalement, les résultats de la méthode lifting restent en dessous de ceux obtenus grâce à l'identification du champ de vecteurs via les différences centrées.

Analysons maintenant les résultats obtenus avec le lissage des données et où l'identification de la dynamique a été réalisée grâce aux différences centrées, ceux-ci se trouvent à la figure 3.10. Comme nous pouvions nous y attendre, les résultats sont meilleurs que pour la méthode lifting. Seul le deuxième réseau de taille 10 identifié via les arbres possède une valeur AUROC en dessous de 0.5, tandis que les meilleurs résultats sont de l'ordre de 0.6. En moyenne, les valeurs AUROC sont supérieures à 0.5.

Comparons ces résultats avec ceux obtenus sans le lissage de données, les valeurs AUROC moyennes sont reprises à la table 3.5. Nous observons que l'identification réalisée après le lissage des données est moins bonne que celle réalisée avec les données initiales. Cette différence est particulièrement marquée lorsque $n = 10$ et que les arbres décisionnels sont utilisés pour la régression.

Ces résultats nous montrent que le lissage de donnée réalisée avec les moyennes

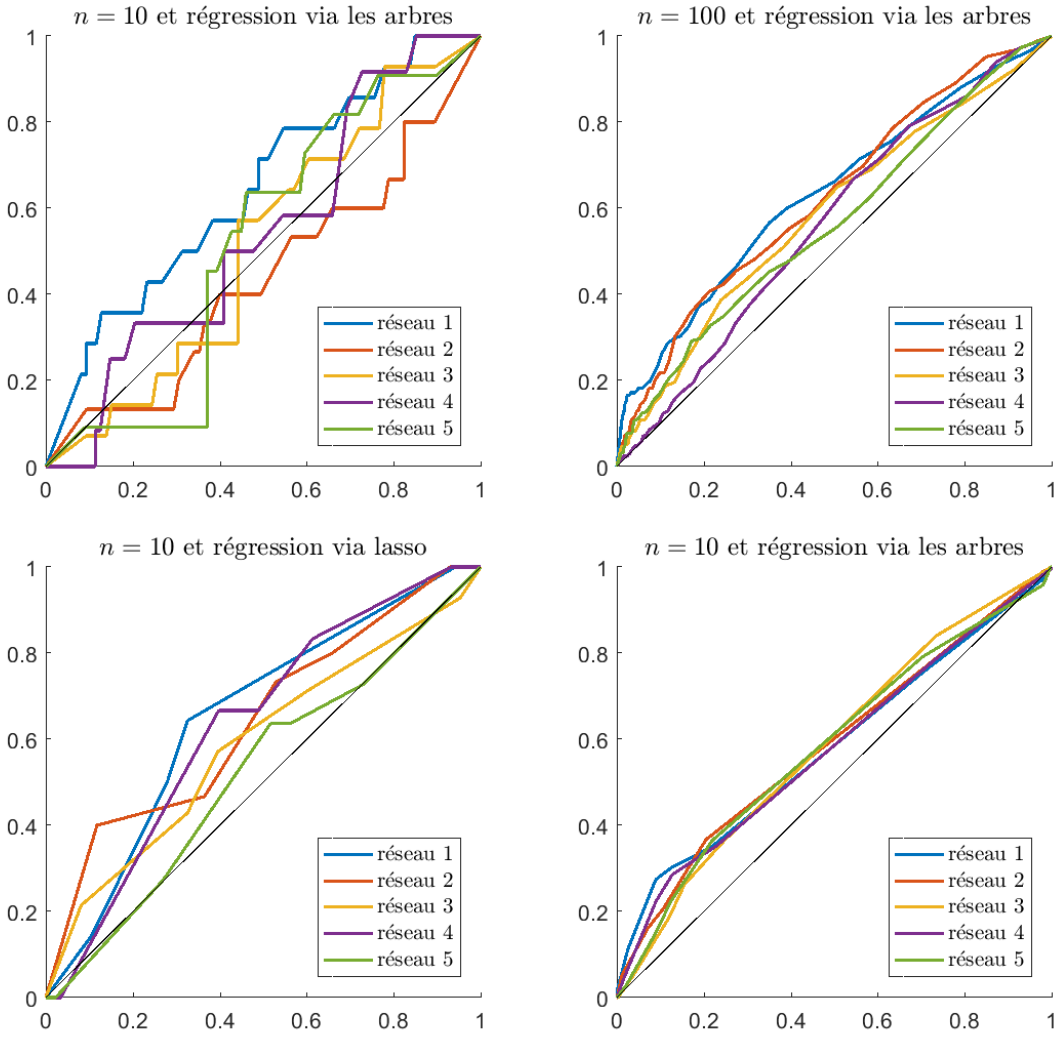


	n=10					n=100				
Arbres	0.49	0.41	0.61	0.6	0.35	0.52	0.53	0.54	0.55	0.46
Lasso	0.69	0.58	0.53	0.49	0.58	0.56	0.56	0.51	0.58	0.52

FIGURE 3.9 – Courbes ROC et valeurs AUROC obtenues en identifiant le champ de vecteur via la méthode lifting pour les données *DREAM-4* après lissage.

	$n = 10$		$n = 100$	
	Sans liss.	Avec liss.	Sans liss.	Avec liss.
Arbres	0.61	0.53	0.65	0.60
Lasso	0.62	0.61	0.62	0.58

TABLE 3.5 – Comparaison des valeurs AUROC moyennes pour les différences centrées avec et sans lissage des données.



	n=10					n=100				
Arbres	0.64	0.44	0.51	0.54	0.52	0.63	0.63	0.59	0.56	0.57
Lasso	0.65	0.64	0.59	0.64	0.52	0.58	0.59	0.59	0.58	0.58

FIGURE 3.10 – Courbes ROC et valeurs AUROC obtenues en identifiant le champ de vecteurs via les différences centrées pour les données *DREAM-4* après lissage.

n_{min} \ D	1	5	10	50	100	250	500
1	0.569	0.581	0.59	0.626	0.629	0.637	0.639
3	0.569	0.601	0.604	0.637	0.638	0.643	0.639
5	0.544	0.601	0.623	0.628	0.634	0.637	0.637
10	0.53	0.602	0.616	0.639	0.636	0.634	0.639
50	0.522	0.569	0.605	0.623	0.629	0.637	0.628

TABLE 3.6 – AUROC moyennes pour l'identification des 10 réseaux en fonction de n_{min} et de D.

mobiles et en fixant $p = 2$ permet d'améliorer légèrement les performances de la méthode lifting avec la régression via le lasso. Pour la méthode avec les différences centrées, les résultats sont moins bons après avoir lissé les données. D'autres tests ont été effectué en fixant $p = 1$ ou $p = 3$ ou encore en utilisant une méthode de lissage comme la régression locale. Néanmoins, cela n'a pas permis d'obtenir de meilleurs résultats.

3.4.4 Amélioration de la régression via les arbres

Nous avons vu que la méthode de régression via les arbres dépendaient de 2 paramètres : D et n_{min}, ces paramètres ont été fixés sur base de l'analyse réalisée dans l'article [6], mais dans cette section, nous chercherons à vérifier si une autre configuration de ces paramètres pourrait être meilleure.

Pour cela, l'identification de réseaux a été à nouveau réalisée sur les 10 réseaux en utilisant les différences centrées pour le calcul du champ de vecteurs afin de pouvoir comparer les performances. Les valeurs possibles de n_{min} sont $1, 3, 5, 10$ et 50 pour D, $1, 5, 10, 50, 100, 250$ ou 500. Pour chacune des combinaisons, les moyennes des valeurs AUROC pour les réseaux ont été calculées et se trouvent à la figure 3.6.

Commençons par analyser l'influence du paramètre D. Comme nous pouvions nous y attendre, augmenter D permet d'améliorer les résultats. L'effet de l'augmentation du nombre d'arbres est surtout visible lorsque ce dernier est petit. En effet, la différence entre les résultats obtenus avec 1 et 5 arbres et plus marquée qu'entre 50 et 100 arbres. Cette croissance n'est pas complètement régulière. Par exemple pour le cas où $n_{min} = 10$, les résultats avec $D = 50$ et $D = 500$ sont les meilleurs, tandis que pour un nombre d'arbres intermédiaires les valeurs AUROC moyennes sont légèrement moins élevées.

Concernant l'influence de n_{min}, son effet sur les résultats est plus marqué lorsque D est petit. En effet, dans ce cas, la régression réalisée doit être la plus précise possible puisque nous ne pouvons pas compter sur un résultat moyen. Lorsque D est grand, les résultats sont issus de moyennes et nous observons moins l'influence du paramètre n_{min}. Nous observons que diminuer n_{min} permet d'augmenter les performances sauf lorsque $n_{min} = 1$. Ces résultats se rapprochent de ceux obtenus dans l'article [6] : diminuer n_{min}

est préférable pour préciser la régression jusqu'à un certain stade.

Au vu de ces résultats, le choix de $n_{min} = 5$ et $D = 100$ paraît approprié. Il aurait été possible d'augmenter D, mais cela augmente considérablement le temps de calcul pour des performances plutôt équivalentes.

3.5 Conclusion et discussion

Nous avons donc utilisé la méthode d'identification comme décrite au chapitre 2 pour identifier les réseaux proposés dans le cadre du concours *DREAM-4*. Nous avons également comparé l'identification basée sur l'opérateur de Koopman avec une identification pour laquelle le champ de vecteurs est calculé au moyen des différences centrées. Nous avons ensuite proposé des pistes d'amélioration pour la méthode. La synthèse des valeurs AUROC obtenues pour les variantes du calcul du champ de vecteurs se trouvent à la figure 3.11.

Les meilleures performances sont obtenues lorsque la dynamique est identifiée grâce aux différences centrées. Celle-ci plus performante lorsque les données sont utilisées de manière brute. Malgré les différentes variantes proposées pour la méthode lifting, aucune ne permet d'obtenir de meilleur résultats que ceux obtenus avec les différences centrées. Notons que lorsque $n = 10$ la régression via lasso est soit plus performante soit équivalente à la régression via les arbres. Lorsque $n = 100$, la méthode lasso est plus efficace uniquement dans le cas où les données ont été lissées.

Un autre paramètre à prendre en compte est la variabilité de la méthode en fonction des différents réseaux. Au travers des différents résultats, nous avons observé que globalement, la régression via le lasso dépendait moins fortement du réseau spécifique que la régression via les arbres décisionnels. Cette différence s'explique notamment par le fait que l'algorithme basé sur les arbres contient une part d'aléatoire et donc les résultats varient.

Nous avons vu dans le chapitre 2 un résultat de convergence théorique pour l'estimation du champ de vecteurs via l'opérateur de Koopman. Force est de constater que dans la pratique, nous n'observons pas des résultats performants. En effet, les valeurs AUROC moyennes n'atteignent pas 0.6 pour la méthode lifting. Essayons d'expliquer ces résultats. Premièrement, ce résultat théorique n'est valable que lorsque que les données sont non bruitées, ce qui n'est pas le cas en pratique. De plus, il se peut que le pas de temps soit trop grand que pour obtenir une bonne approximation. Nous sommes également limités par K, le nombre de paires de données, lui-même fixé par le nombre de données temporelles. À cela s'ajoute le fait que nous n'avons que des données partielles du systèmes. En effet, les données sont construites via des équations différentielles exprimant la concentrations en ARNm, mais aussi en protéines. Or, seuls le taux d'ADN est utilisé pour inférer le réseau. Finalement, il reste le problème du calcul effectif du logarithme matriciel. Dans la pratique, celui-ci est complexe et nous négligeons donc sa partie imaginaire. Cela entraîne forcément des erreurs.

$n = 10$						
Identification de F	lifting duale	différences centrées	lifting modif.	lifting + lissage	diff. centrées + lissage	lifting main
Arbres	0.54	0.61	0.53	0.49	0.53	0.48
Lasso	0.54	0.61	0.53	0.57	0.61	0.58

$n = 100$						
Identification de F	lifting dual	différences centrées	lifting modif.	lifting + lissage	diff. centrées + lissage	lifting main
Arbres	0.52	0.65	0.52	0.52	0.60	×
Lasso	0.52	0.62	0.51	0.55	0.58	×

FIGURE 3.11 – Synthèse des valeurs AUROC moyennes pour les différentes configurations

Notons que les données du concours sont de meilleure qualité et en plus grande quantité que les données habituellement disponibles pour des réseaux réels [19]. De plus les réseaux utilisés ne sont qu'une simplification des réseaux réels. Par exemple, ils ne tiennent pas compte de la régulation post-transcriptionnelle [24]. Malgré cela les performances obtenus sur les réseaux synthétiques *DREAM-4* ne sont que légèrement meilleures qu'une identification aléatoire.

Les meilleurs valeurs AUROC obtenues sont de l'ordre de 0.6 grâce à l'identification du champ de vecteurs au moyen des différences centrées. Évaluons ce résultat par rapport au contexte du concours *DREAM-4*. Une majorité des réseaux inférés par les participants donnaient lieu à des résultats équivalents à une identification aléatoire [3]. Pourtant les méthodes d'identification étaient variées et certaines avaient également été publiées et validées de manière expérimentales [24]. Cette différence de résultats peut notamment s'expliquer par le fait que les méthodes sont testées sur des réseaux spécifiques et en général plus simples. Nous ne pouvons pas attendre les même performances pour les données *DREAM-4*. Obtenir des résultats significativement supérieurs aux performances d'une identification aléatoire est déjà un point positif, même si ceux-ci restent faibles.

L'étude des performances d'une méthode d'identification sur des réseaux synthétiques ne remplacent pas l'étude des performances avec des réseaux réels. Néanmoins, ces données synthétiques constituent un bon test pour avoir une idée de l'efficacité d'une méthode. Notons tout de même que les performances dépendent fortement du réseau et qu'elles sont étudiées pour des réseaux spécifiques. Cela signifie qu'il n'est pas possible de les généraliser pour d'autres réseaux dont la structure est différente ou inconnue [24].

CHAPITRE 4

IDENTIFICATION DU RÉSEAU DE L'HORLOGE CIRCADIENNE DE L'*ARABIDOPSIS THALIANA*

Dans ce chapitre, nous allons identifier le réseaux de régulation génétique de la plante Arabidopsis thaliana grâce à la méthode développée au chapitre 2. Nous nous intéresserons plus particulièrement à son horloge circadienne. Afin de pouvoir analyser les résultats, nous les comparerons avec ceux d'une autre étude.

4.1 Contexte

Nous allons inférer un réseau de régulation génétique sur base de données réelles. Nous travaillerons avec la plante Arabidopsis thaliana couramment appelée Arabette des dames. Cette plante est largement étudiée dans le domaine de la biologie [26]. En effet, celle-ci présente plusieurs avantages pour la recherche : elle a un cycle de vie rapide (6 semaines) et elle est facile à cultiver. De plus, elle possède un génome relativement petit, ce qui a permis aux chercheurs de le séquencer complètement dès les années 2000.

Malgré que son génome soit relativement petit par rapport à d'autre organismes, elle contient tout de même environ 23 000 gènes. Rappelons que les gènes sont des parties du génome codant (c'est-à-dire qui permettent de synthétiser des protéines). Nous n'allons donc travailler qu'avec une faible partie de ces gènes et pour cela nous sélectionnerons les gènes intervenant dans l'horloge circadienne. Celle-ci est un mécanisme interne qui intervient dans le contrôle des rythmes quotidiens. Elle permet notamment de s'adapter aux changements extérieurs (variation de la température, cycle jour/nuit, etc). Déterminer la dynamique de cette horloge permet de mieux comprendre le mécanisme de l'activité biologique quotidienne.

Les données utilisées proviennent de l'article [2]. Elles ont été obtenues de la manière suivante. Différentes cultures ont été lancées en même temps. Chaque plante s'est développée selon les mêmes conditions. À partir du onzième jour, des mesures ont été effectuées via des puces à ADN pour obtenir le taux d'ARNm dans les gènes. Les me-

sures ont été réalisées toutes les 4h et ce pendant 48h. Cela donne une série temporelle de 12 points. Les données mises à disposition proviennent de moyenne entre plusieurs plantes. Pour une partie de celle-ci, la substance nicotinamide a été administrée toutes les deux heures pour permettre d'étudier la réaction de la plante face à cette perturbation. Les recherches précédentes ont montré que cette substance ralentissait le rythme de l'horloge [18]. Dans un premier temps, nous nous concentrerons sur le réseau sans perturbation.

Commençons par identifier les gènes qui interviennent dans l'horloge circadienne. Nous savons que la régulation de ces gènes doit ressembler à une sinusoïde d'une période d'environ 24h. Les auteurs des articles [2] et [18] ont proposé la méthode suivante. Ils commencent par sélectionner les gènes reconnus intervenir dans l'horloge circadienne. Ils approximent les différentes courbes du taux d'ARNm de ces gènes par une fonction pseudo-sinusoïdale. Finalement, ils sélectionnent celles dont la probabilité d'être périodique est importante.

Au final, 11 gènes ont été sélectionnés et ce sont ceux-ci que nous utiliserons : PRR9, PRR7, PRR5, LHY, CCA1, TOC1, GI, ELF4, CRY2, PHYA et RVE8. Les taux d'ARNm mesurés dans ces gènes sont représentés à la figure 4.1. Pour le cas non traité, les données semblent effectivement montrer une périodicité d'environ 24h. Ce phénomène est particulièrement visible pour les gènes GI ou CCA1. Concernant l'influence de la substance, celle-ci semble différente en fonction des gènes. Pour les gènes TOC1, GI et ELF4, la présence de nicotinamide semble retarder la système. Le traitement agit également en termes quantitatifs : pour le gène PHYA, le taux d'ARNm est considérablement plus important tandis que pour RVE8, il semble stable à un niveau bas. Dans la pratique, ces données sont normalisées. En effet, nous observons que les échelles du taux d'ARNm sont fort différentes en fonction des gènes.

4.2 Comparaison

Comme le réseau de régulation réel de l'horloge circadienne de la plante Arabidopsis thaliana n'est pas connu, le seul moyen d'évaluer les résultats est de les comparer avec ceux d'autres études. Pour cela nous utiliserons les références [17] et [18] dont les auteurs ont travaillé avec les mêmes données. Nous allons présenter brièvement leur méthode d'identification.

Pour modéliser le réseau, ils considèrent des modèles pour chacune des paires de gènes possibles. Comme il y a 11 gènes et que nous considérons qu'un gène ne s'autorégule pas, il y a donc 110 possibilités. Ces modèles sont linéaires et invariants par rapport au temps. Ils sont de la forme suivante :

$$\frac{dv(t)}{dt} = au(t) - bv(t) + c$$

où $u(t)$ et $v(t)$ représente les taux d'ARNm dans le gène régulateur et le gène régulé. Les paramètres a, b et c sont identifiés de manière à minimiser l'erreur lors de l'estimation de $v(t)$. Si cette erreur est en-dessous d'un seuil, alors nous considérons que le gène u

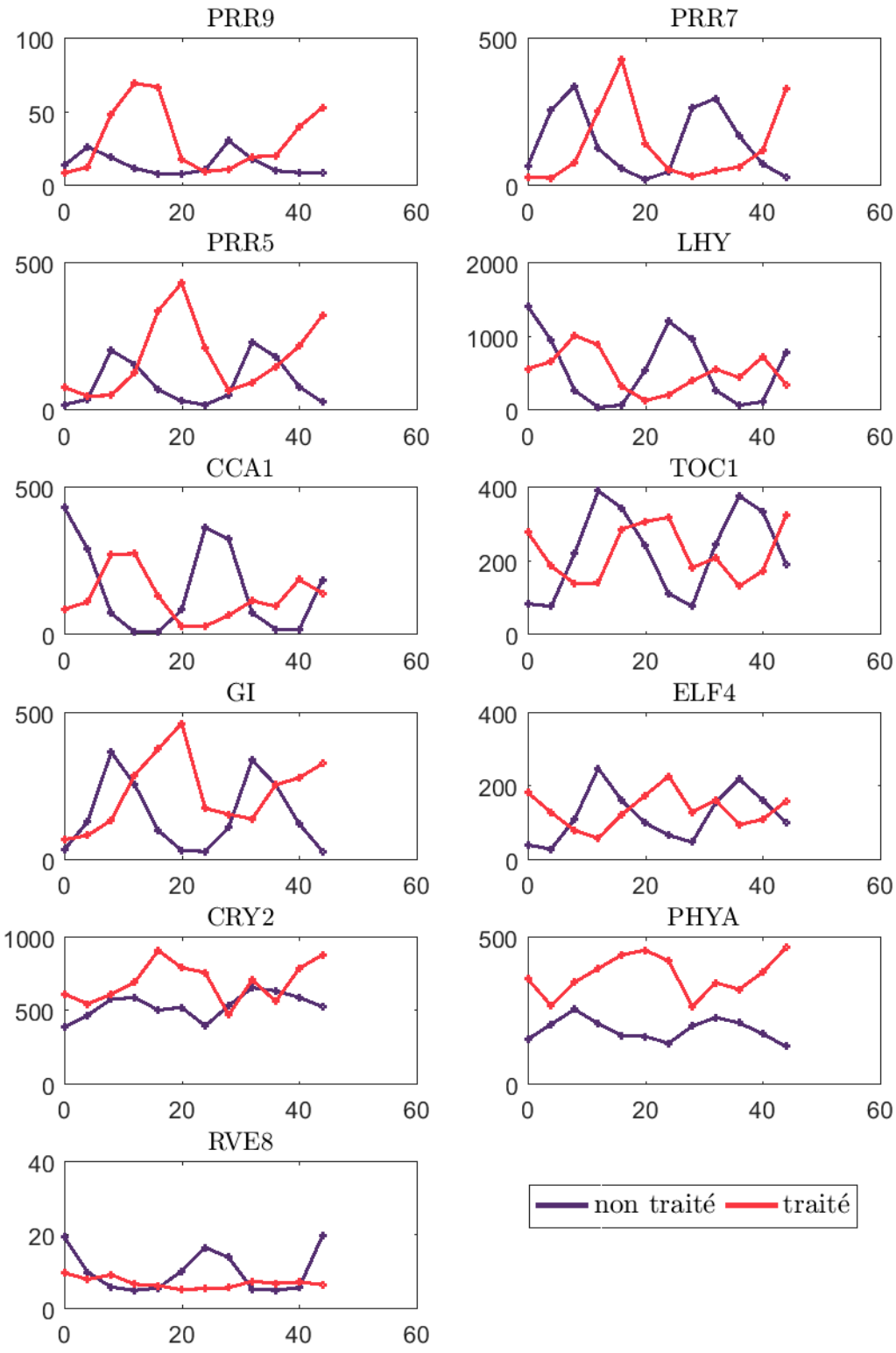


FIGURE 4.1 – Taux d'ARNM dans les 11 gène impliqués dans l'horloge circadienne

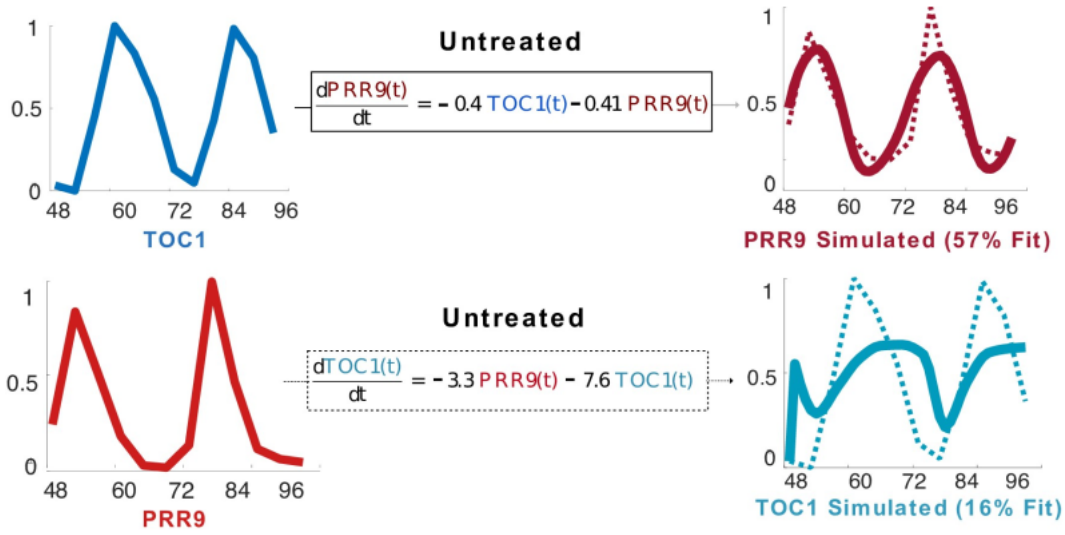


FIGURE 4.2 – Illustration de l'identification des liens selon l'article [17].

influence le gène v. Cette procédure est illustrée à la figure 4.2 avec pour exemple les gènes PRR9 et TOC1. Le premier modèle vise à expliquer l'évolution du taux d'ARNm dans le gène PRR9 en fonction du gène TOC1. Le modèle ainsi construit correspond à 57% aux mesures. Le seuil est fixé à 46%, cela signifie que le modèle considère que le gène TOC1 influence PRR9. Au contraire, l'évolution du gène PRR9 permet d'expliquer que 16% du comportement du taux d'ARNm contenu dans le gène TOC1, le modèle considère donc que l'influence n'est pas bilatérale. Cette méthode d'identification de réseau est désignée par All-to-all (ATA).

Dans un deuxième temps, l'objectif est d'identifier les gènes qui sont influencés par la présence de la substance nicotinamide. Tous les modèles validés pour la situation sans traitement (c'est-à-dire ceux pouvant expliquer au moins 46% des données) sont à nouveau estimés pour le cas traité. Si ces nouveaux modèles correspondent toujours au moins à 46% aux données avec présence de nicotinamide, alors le modèle est validé pour le cas non traité. Cette procédure implique qu'il y aura plus de liens identifiés pour le cas sans traitement plutôt qu'avec.

Les modèles estimés sans et avec traitement sont ensuite comparés. S'ils différent fortement, cela signifie que la substance a eu une influence sur le gène régulateur du modèle. Pour comparer les modèles, une métrique $\nu - gap$ est utilisée. Celle-ci mesure la distance entre deux modèles linéaires et attribue une valeur entre 0 et 1. Soit deux modèles m_1 et m_2 correspondant respectivement à la situation sans et avec traitement dont la distance $\nu - gap$ est supérieure à 0.2, alors le gène régulateur des modèles m_1 et m_2 est considéré comme un point d'entrée pour la substance introduite.

Cette procédure a l'avantage de pouvoir comparer les modèles avant et après et ainsi espérer détecter les gènes fortement influencés par la présence de nicotinamide. Néanmoins, elle recherche uniquement comment les liens déjà existants sont modifiés.

Le modèle avec traitement contient d'office moins de liens que celui sans traitement. Nous pourrions également rechercher si l'introduction de traitement en provoque pas l'apparition de nouveaux liens. De plus, ce modèle présente deux désavantages. Tout d'abord, il suppose que les gènes interagissent entre eux de manière linéaire, or ce n'est pas le cas. Deuxièmement, les liens entre les gènes sont identifiés deux à deux et non globalement contrairement à la méthode lifting.

4.3 Procédure

Nous allons donc appliquer la méthode d'identification de réseaux comme proposé au chapitre 2 sur les données Arabidopsis thaliana. Nous avons le choix

— de la méthode pour calculer le champ de vecteurs (via lifting ou les différences centrées) ;

— de la régression (méthode basée sur les arbres décisionnels ou le lasso).

Cela permet au total d'identifier le réseau selon 4 variantes. Néanmoins, nous allons nous concentrer uniquement sur la régression via le lasso. En effet, nous avons observé que la régression via les arbres était irrégulière puisqu'elle contient une part de hasard. De plus le nombre d'observations dans notre cas est faible ($K = 11$), or son efficacité a été montrée uniquement pour des jeux de données contenant un nombre largement supérieur d'observations [6].

Il faut également choisir quelle variante de la méthode lifting appliquer. Nous avons le nombre de gènes $n = 11$ et le nombre de mesures temporelles $\tilde{K} = 12$. Il est donc possible de créer $K = 11$ paires de données utilisables pour la méthode lifting. Si nous utilisons la variante principale, nous sommes limités par le nombre de données et devons exprimer la dynamique uniquement en fonction des monômes de degré 1. Ce choix n'est pas judicieux, c'est pourquoi nous utiliserons la variante duale qui s'applique lorsqu'il y a peu de données disponibles. Le choix du paramètre $\gamma = 0.01$ est conservé.

Nous avons choisi d'identifier le champ de vecteurs uniquement sur les 11 gènes étudiés, mais il aurait été également envisageable d'identifier la dynamique du système sur un nombre plus large de gènes. Nous aurions pu imaginer évaluer la dynamique de l'ensemble des gènes soit de 22810 éléments. Dans ce cas, il faut fixer une valeur de γ largement inférieure. En effet, le choix de γ dépend de la dimension du système. Pour rappel, la paramètre γ intervient dans la définition de la fonction g_k

$$g_k(x) = e^{-\gamma||x-x_k||^2}.$$

Cela signifie que lorsque la dimension de x augmente, la norme de $||x - x_k||$ augmente également. Pour éviter d'avoir l'évaluation de $g_k(x)$ numériquement arrondie à 0, il faut que γ soit suffisamment petit. Pour un bon choix de γ, la méthode fonctionne numériquement et permet d'obtenir le champ de vecteurs. Néanmoins nous n'avons pas de garantie sur ce résultat. Bien que la méthode lifting soit efficace pour des petits jeux de données, elle n'a pas été testée pour des systèmes de dimension aussi grande. De plus, cela a du sens de ne considérer uniquement les gènes impliqués dans l'horloge circadienne puisque nous pouvons supposer qu'ils s'influencent plus entre eux qu'avec

des gènes remplissant d'autres fonctions. Il aurait été également possible de sélectionner une partie des gènes plus grande que 11 éléments mais sans tous les considérer. Pour que ce choix soit judicieux, il faudrait avoir des connaissance a priori sur le système et sélectionner les gènes supposés interagir avec ceux de l'horloge circadienne.

4.4 Résultats

4.4.1 Données non traitées

L'identification de réseau a été réalisée pour les données non traitées. Le paramètre λ a été choisi de manière à avoir approximativement le même nombre de liens que pour le réseau identifié selon la méthode ATA afin de pouvoir comparer les résultats. Leur modèle contenaient au total 68 liens pour le cas non traité. Avec le choix du paramètre $\lambda = 0.001$, 67 liens sont identifiés pour la méthode lifting et 62 pour les différences centrées.

Pour représenter ces données, nous n'allons pas utiliser le graphe complet, mais plutôt nous intéresser à la connectivité. En effet, l'objectif est de rechercher quels sont les gènes qui influencent le plus le modèle, nous aurons donc une approche plus globale. Nous utiliserons le degré de chaque gène, c'est-à-dire le nombre de liens sortants. Les résultats obtenus se trouvent à la figure 4.3. À titre de comparaison, les résultats obtenus avec la méthode ATA sont indiqués en vert [17].

Les 3 identifications de réseaux s'accordent pour dire que le gène PRR9 joue un rôle important dans le fonctionnement de l'horloge circadienne puisqu'il influence entre 8 et 10 autres gènes. Concernant les gènes PRR7, PRR5 et GI, ils jouent également un rôle important dans le modèle ATA, mais selon les méthodes lifting et différences centrées, leur importance est moindre. Au contraire, les degrés des gènes ELF4, CRY2 et PHYA estimés via la méthode lifting sont supérieurs aux résultats de la référence. C'est également le cas pour l'identification effectuée avec les différences centrées pour le gène ELF4. Finalement, pour le reste des gènes, les résultats obtenus avec les méthodes lifting et différences centrées sont similaires à ceux obtenus avec ATA. Entre l'identification via lifting et via les différences centrées, aucune des deux n'est plus proche que l'autre de la méthode ATA.

Notons que la comparaison est réalisée à titre indicatif, mais nous ne connaissons pas le réseau réel afin de pouvoir comparer les résultats. Il n'y a pas de garantie que la méthode ATA soit plus efficace que la méthode lifting ou des différences centrées. Par ailleurs, la méthode ATA a également été testée sur les données inspirées du concours *DREAM-4 in silico network challenge* [16]. Leur résultats ne sont pas tout à fait comparables avec ce qui a été fait au chapitre 3. En effet, les auteurs de l'article [16] ont généré des données selon un modèle similaire mais pour lequel une ou des perturbation(s) ont été ajoutées. Dans le cas d'une perturbation unique et pour des réseaux de taille 10 les valeurs AUROC calculées via la méthode ATA se situent entre 0.5 et 0.6. Autrement dit, les performances de la méthode ATA sur des données synthétiques sont du même ordre que pour la méthode lifting.

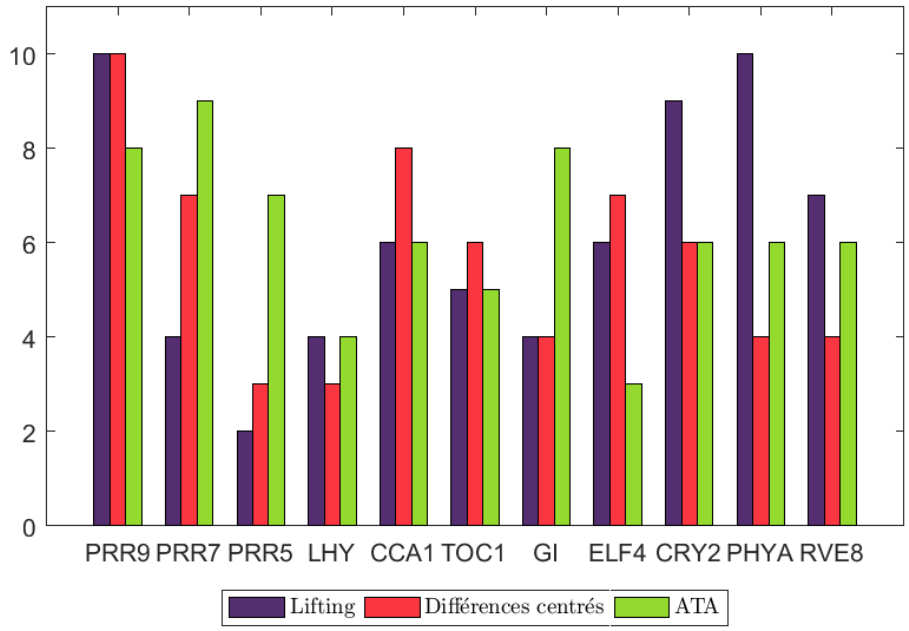


FIGURE 4.3 – Degré des 11 gènes de l'horloge circadienne de l'Arabidopsis thaliana pour le cas non traité.

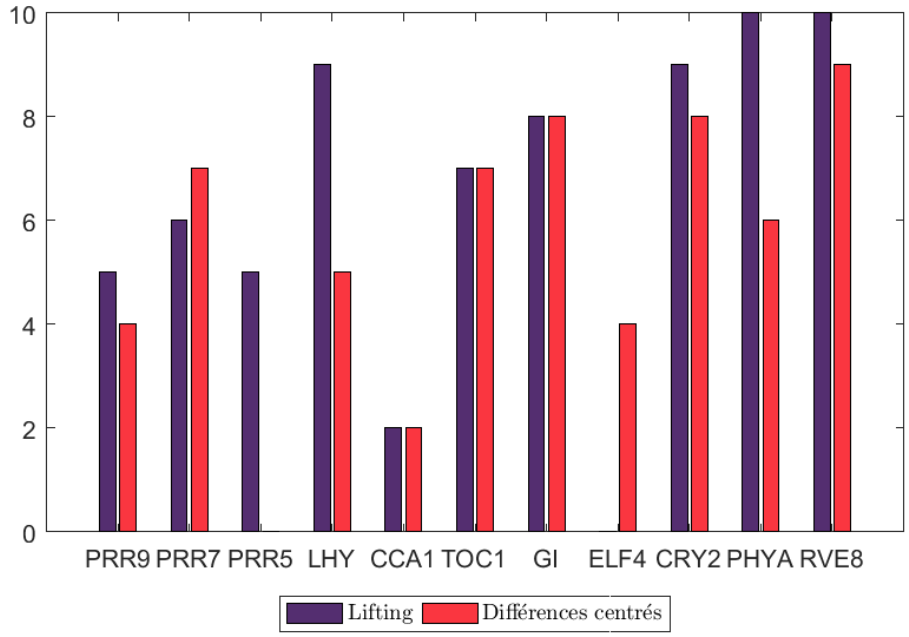


FIGURE 4.4 – Degré des 11 gènes de l'horloge circadienne de l'Arabidopsis thaliana pour la cas traité.

4.4.2 Analyse de l'influence du traitement

L'identification de réseau a été à nouveau réalisée pour les données non traitées. Les degrés des gènes se trouvent à la figure 4.4. Cette fois-ci il n'y a pas les résultats de la méthode ATA pour comparer. Comme expliqué précédemment, les auteurs de [17], ne réévalue que les liens existants déjà dans le modèle non traité et ne permettent donc pas l'apparition de nouveaux liens. Le modèle obtenu ne peut donc pas être comparé avec les résultats des méthodes lifting et différences centrées.

Le premier constat que nous pouvons faire, c'est que certains gènes semblent ne plus avoir d'influence sur les autres : PRR5 pour la méthode lifting et ELF4 pour les différences centrées. Ensuite, le gène PRR9 semble avoir une importance moindre. Les gènes GI,CRY2, PHYA et RVE8 sont eux davantage connectés aux autres. C'est également la cas du gène LHY pour la méthode lifting.

Les résultats avec la méthode ATA montraient que les gènes CCA1, TOC1 et PRR7 étaient ceux qui perdaient le plus de connectivité. Dans notre modèle, nous vérifions également que le gène CCA1 est moins connecté suite à l'introduction de la substance. Néanmoins pour les deux autres, nous observons l'effet inverse : les gènes sont plus connectés.

4.5 Discussion

Le réseau de l'horloge circadienne de la plante Arabidopsis thaliana a pu être identifié grâce à la méthode développée au chapitre 2. Ces résultats ont été comparés avec ceux obtenus via la méthode ATA. Néanmoins, il n'est pas possible de vérifier les résultats obtenus puisque le réseau réel reste inconnu. Au vu des résultats obtenus pour les différentes méthodes pour des données synthétiques, il n'est pas possible de pouvoir assurer que l'identification de ces réseaux est satisfaisante. Pour pouvoir valider ou non les réseaux obtenus, il serait possible de comparer ces résultats avec d'autres modèles ou encore de réaliser une analyse en tenant compte des aspects biologiques déjà établis.

Nous avons également essayé de déterminer quels étaient les gènes fortement influencés par la présence de nicotinamide. La simplification sous forme de graphe rend difficile l'analyse de cette influence tandis que la méthode ATA propose de comparer les modèles obtenus pour les interactions entre chaque paire de gènes. Elle est donc plus adaptée à l'étude de l'influence de facteur ajouté au système.

CONCLUSION

Dans ce mémoire, nous avons présenté une nouvelle méthode d'identification de réseau basée sur l'opérateur de Koopman. Celle-ci identifie la dynamique d'un système en transférant les données dans un nouvel espace de fonctions $\mathcal{F}$. Au sein de cet espace, la dynamique est linéaire et donc facile à identifier. Ensuite le champ de vecteurs est évalué dans l'espace de départ grâce à la dynamique observée dans l'espace $\mathcal{F}$. Des résultats théoriques garantissent l'efficacité de la méthode dans un cas idéal où les données ne sont pas bruitées. De plus, cette méthode a été testée sur des exemples synthétiques et montrent de bons résultats comme illustré à la section 2.1.6. Néanmoins, il serait intéressant d'évaluer la méthode sur des données synthétiques non bruitées ou faiblement, avec des systèmes de plus grandes dimensions pour rechercher ses limites.

La méthode lifting présente deux avantages majeurs. Tout d'abord, bien qu'elle utilise des méthodes linéaires, elle est valide pour des système dont la dynamique est non linéaire. Ensuite il existe une variante duale pour laquelle le champ de vecteurs peut être estimé lorsque le nombres de paramètres à inférer N est plus grand que le nombre d'observations K.

Ces avantages sont particulièrement intéressants dans le contexte des réseaux de régulation génétique. En effet, ceux-ci contiennent en général peu d'observations et beaucoup de paramètres. La variante duale semble donc être une bonne alternative aux méthodes proposées habituellement. De plus l'identification via lifting peut être appliquée pour des systèmes non linéaires, ce qui est le cas des réseaux de régulation génétique.

L'identification du champ de vecteurs a été réalisée pour les données synthétiques du concours *DREAM-4 in silico network challenge*. Ces données ont été construites de manière à ressembler au problème réel. Bien que théoriquement la méthode soit performante, les résultats ne sont que peu satisfaisants. En effet, les valeurs AUROC moyennes pour la méthode lifting sont situées entre 0.48 et 0.58. Celles-ci restent en-dessous de l'identification réalisée avec les différences centrées. Cela peut s'expliquer par plusieurs phénomènes : le bruit des données, l'observation partielle ou encore par le problème du calcul du logarithme matriciel comme discuté à la section 3.5.

Notons également, que même si les données synthétiques sont en plus grande quantité que les données réelles, le problème d'inférence de réseau de régulation génétique reste complexe. Le nombre d'éléments dans le système est important et nous utilisons uniquement la concentration en ARNm et pas celle en protéines. Par ailleurs, les résultats du concours montrent qu'une grande partie des méthode d'identification employées ne sont pas satisfaisantes. De plus, pour la version 3 de ce concours, il a été montré que les 5 meilleurs méthodes d'identification de réseaux étaient basées sur des approches différentes [24]. Il est donc difficile de pouvoir déterminer quelles sont les stratégies les plus efficaces. Il est possible qu'en appliquant cette méthode à d'autres réseaux que ceux issus de la génétique, les performances soient meilleures. Il serait alors intéressant de comparer la méthode lifting entre différents types de réseaux.

Finalement nous avons identifié le réseau des gènes intervenant dans l'horloge circadienne de la plante Arbidopsis thaliana. Nous avons comparé ce réseau avec les résultats obtenus dans les références [16] et [18]. Néanmoins, nous n'avons pas la possibilité de pouvoir valider ou non ce modèle. Dans un futur travail, nous pourrions comparer le modèle obtenu avec les résultats d'autres études. De plus nous nous sommes concentrés sur l'aspect mathématique du problème, mais nos analyses gagneraient à être appuyées par les résultats de la recherche en biologie.

BIBLIOGRAPHIE

[1] Auliac C., *Approches évolutionnaires pour la reconstruction de réseaux de régulation génétique par apprentissage de réseaux bayésiens*, thèse, Université d'Evry-Val d'Essonne, France 2008.

[2] Dalchau N, Hubbard KE, Robertson FC, Hotta CT, Briggs HM, Stan GB, Gonçalves JM, Webb AA, *Correct biological timing in Arabidopsis requires multiple light-signaling pathways*, Proc Natl Acad Sci U S A 107(29), 2010

[3] *DREAM 4 - In Silico Network Chalenge*, www.synapse.org, 2014.

[4] Drulhe Samuel, *Identification de réseaux de régulation génique à partir de données d'expression : une approche basée sur les modèles affines par morceau*, Biophysique, Université Jospeh-Fourier, Grenoble, 2008.

[5] Frölich F., Loos C., Haseneauer J., *Scalable inference of ordinary differentiel equation models of biochimical processes*, Gene regulatory networks : methods and protocols, NY : Springer New Yprk, p. 385-422, 2019.

[6] Geurts P., Ernst D., Wehenkel L., *Extremely randomized trees*, Machine learning, 36, p.3-42, 2006.

[7] Geurts P., *Regression tree package*, https ://people.montefiore.uliege.be/geurts, 2010.

[8] Geurts P., *Classification and regression trees*, support de cours, Université de Liège, 2019.

[9] Hastie T., Tibshirani R., Friedman J., *The elements of statistical learning : data mining, inference and prediction*, seconde édition, Springer,2017.

[10] Hyndman R.J., *Moving averages*, 2009.

[11] Higham N.J., *Functions of matrices, theory and computation*, Society for industrial and applied mathematics, 2008.

[12] Huynh-Thu V.A., Sanguinetti G., *Gene Regulatory Networks : an introduction survey methods and protocols*, Methods in molecular biology, 1-23, 2019.

[13] Ismaili A. et Gaillard P., *Le lasso, ou comment choisir parmi un grand nombre de variables à l'aide de peu d'observations*, Université de Paris-Sud, 2009.

[14] Karlebach G., Shamir R., *Modelling and analysis of gene regulatory networks*, Nature reviews molecular cell Biology, 9, 770–780, 2008.

[15] Louppe G., Wehenkel L., Sutera A. et Geurts P., *Understanding variable importances in firests of randomized tress*, Advances in Neural Information Processing Systems, 26, 2013.

[16] Mombaerts L., Aalto A., Markdahl J., Gonçalves J., *A multifactorial evaluation framework for gene regulatory network reconstruction*, IFAC (International Federation of Automatic Control), 52 (26), p.262-268, 2019.

[17] Mombaerts L, Carignano A, Robertson FC, Hearn TJ, Junyang J, Hayden D, Rutterford Z, Hotta CT, Hubbard KE, Maria MRC, Yuan Y, Hannah MA, Goncalves J, Webb AAR, *Dynamical differential expression (DyDE) reveals the period control mechanisms of the Arabidopsis circadian oscillator*, PLoS Comput Biol, 15(1), 2019.

[18] Mombaerts L, *Dynamical modeling techniques for biological time series data*, thèse, Université de Luxembourg, Luxembourg, 2019.

[19] Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G. *Revealing strengths and weaknesses of methods for gene network inference*, PNAS, 107(14), p.6286-6291, 2010.

[20] Marbach D, Schaffter T, Mattiussi C, Floreano D., *Generating Realistic in silico Gene Networks for Performance Assessment of Reverse Engineering Methods*, Journal of Computational Biology, 16(2), p. 229-239, 2009.

[21] Mauroy A., Goncalves J., *Koopman-based lifting techniques for nonlinear systems identification*, IEEE Transactions on Automatic Control, 1-16, 2019.

[22] Mauroy A., Goncalves J., *Linear identification of nonlinear systems : A lifting technique based on the Koopman operator*, IEEE 55th Conference on Decision and Control (CDC), 2016.

[23] Penfold C.A., *Wild D.L., How to infer gene networks from expression profiles, revisited*, Interface focus, 1(6), 857-870, 2011.

[24] Prill RJ, Marbach D, Saez-Rodriguez J, Sorger PK, Alexopoulos LG, *Towards a rigorous assessment of systems biology models : The DREAM3 challenges* PLOS ONE, 5(2), 2010.

[25] Ristevski B., *A survey of models for inference of gene regulatory networks*, Nonlinear Analysis : Modelling and Control, 18, 444–465, 2013.

[26] Tair, *About Arabidopsis*, https ://www.arabidopsis.org/index.jsp, consulté en 2021.

[27] Tibshirani R.J., *The lasso problem and uniqueness*, Carnegie Mellon University, 2012.

[28] Thieffry D., De Jong H., *Modélisation, analyse et simulation des réseaux génétiques*, Medecine/sciences, 18, 492-502, 2002.

[29] Zhand Y., Ray S., Guo W., *On the consistency of feature selection with lasso for non-linear targets*, Proceedings of The 33rd International Conference on Machine Learning, New York, p. 183-191, 2016.