UNIVERSITÉ PARIS 13, SORBONNE PARIS CITÉ

ÉCOLE DOCTORALE GALILÉE

THÈSE

Présentée par

Ines Chebil

pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

SPÉCIALITÉ : INFORMATIQUE

Méthodes d'ensemble pour l'inférence de réseaux de régulation coopératifs

soutenue publiquement Septembre 2014

Membre du jury :

Directeurs de thèse :		
	Céline ROUVEIROL	LIPN, Université Paris 13
Président :		
	Younés BENNANI	LIPN, Université Paris 13
Rapporteurs :		
	Florence D'ALCHÉ-BUC	IBISC, Université d'Évry
	Jérôme AZÉ	LIRMM, Université Montpellier 2
Examinateurs :		
	Antoine CORNUÉJOLS	LRI, AgroParisTech

Abstract

Reconstruction of Gene Regulatory Networks (GRNs) is an important step towards understanding the complex regulatory mechanisms within the cell. Many modeling approaches have been introduced to find the causal relationship between genes using expression data. However, they suffer from the high dimensionality problem *i.e.*, having a large number of genes but a small number of samples negatively impacts the results. Moreover, these models do not truthfully reflect the biological system where genes interactions are performed in a cooperative manner but rather simplify the problem by tackling only binary interactions. In this thesis, we present new methods for cooperative GRN inference to improve the stability and accuracy of GRNs reconstruction leveraging ensemble methods. For a given target gene, we extract an ensemble of GRNs from discretized expression data. Inferred networks are then evaluated by ranking individual regulation relationships using a regression based technique and continuous expression data. Evaluations on DREAM5 challenge data as well as human cancer data demonstrate that our methods are efficient, especially when operating on a small data set.

Résumé

La reconstruction des réseaux de régulation génétique (GRNs) est une étape importante pour la compréhension des mécanismes de régulation complexes régissant le fonctionnement de la cellule. De nombreuses approches de modélisation ont été introduites pour inférer le lien de causalité entre les gènes à l'aide des données d'expression génétiques. Cependant, les performances de ces approches sont limitées principalement à cause à des données de grande dimension. En plus, ces méthodes ne reflètent pas généralement la réalité biologique où l'interaction entre gène est réalisée d'une manière coopérative mais considèrent un modèle plus simple où seules les interactions binaires sont considérées. Dans cette thèse, nous présentons de nouvelles méthodes d'inférence de GRN coopératifs afin d'améliorer la stabilité et la précision de la reconstruction des GRNs en utilisant des techniques d'ensemble. Pour un gène cible donné, nous extrayons un ensemble de GRNs coopératifs à partir de données discrétisées d'expression. Les GRNs ainsi que les interactions génétiques inférés sont classés selon leur importance en utilisant la régression linéaire sur la base des données d'expression continues. Les évaluations menées sur les données du challenge DREAM5 et sur des données humaines de cancer de la vessie démontrent que nos méthodes sont efficaces, tout particulièrement si la taille des données d'apprentissage est petite.

Table des matières

Introduction

Le vingtième siècle a connu de nombreuses découvertes qui ont fait des sciences du vivant l'un des domaines les plus productifs en terme de recherche scientifique et technologique. Depuis la découverte du premier modèle de la structure de la molécule d'*acide désoxyribo-nucléique* (ADN) par Watson et Crick en 1953, le chemin parcouru est immense. Moins de 40 années plus tard, en 1990, débutait le projet mondial HGP [1] de séquençage du génome humain. Achevé en 2003, ce projet, appuyé par quatre décennies de progrès technologiques, a permis de constituer une base de données gigantesque référençant l'ensemble des gènes de l'espèce humaine. Une autre avancée technique en biologie sont les *technologies à haut débit* qui permettent, par exemple, de visualiser à un instant donné le niveau d'expression d'un ensemble de plusieurs milliers de gènes (voir annexe A).

L'utilisation de ces techniques a notamment permis de générer de multiples bases de données dont le traitement permettra à terme de nombreuses applications dans des domaines aussi variés que l'agro-alimentaire, la pharmacologie et la médecine.

Grâce aux nombreux travaux entrepris depuis environ un demi-siècle, la somme des connaissances en biologie cellulaire s'est donc vue accroître de manière phénoménale ouvrant la voie à un nouveau domaine pluridisciplinaire, *la biologie des systèmes*.

Les principaux enjeux de cette discipline en pleine expansion sont de modéliser, identifier et éventuellement simuler les réseaux d'interactions entre molécules qui interviennent à différents niveaux dans la cellule. Les systèmes biologiques complexes sont naturellement représentés comme des réseaux d'interactions entre leurs différents composants [BdlFM02, AA03]. Quatre types de réseaux d'interactions moléculaires ont été l'objet de la plupart des études : les réseaux d'interaction protéine-protéine, les réseaux métaboliques, les réseaux de transduction du signal, et les réseaux de régulation génétique. Dans notre travail, nous nous intéressons aux *réseaux de régulation génétique*, qui jouent un rôle fondamental dans le contrôle du fonctionnement et du développement des organismes vivants.

Nous pouvons distinguer deux principaux axes de recherche pour l'étude de ces réseaux : l'*inférence*, en général par des méthodes d'apprentissage automatique ou statistique, de réseaux à partir de données produites par des techniques à haut débit et l'*analyse*, en général par modélisation et simulation, des propriétés dynamiques de réseaux. Ces deux approches sont complémentaires, puisque leurs objectifs sont respectivement d'expliciter la structure des réseaux à partir de données, et de modéliser et simuler leur fonctionnement à partir de cette structure. Nous nous intéressons, dans le cadre de cette thèse, au problème de l'inférence de réseaux de régulation génétique à partir de données. Les fondements de ces recherches ont été posés dans les années 60 par les chercheurs A. Lwoff, F. Jacob et J. Monod [JM61] qui ont décrit pour la première fois un ensemble de faits biologiques permettant d'imaginer une structure de régulation de l'expression de certains gènes de la bactérie *Escherichia coli*. Toutefois, même si la notion de régulation génétique reste un champ de recherche classique en biologie cellulaire, la modélisation de ce phénomène demeure un problème épineux.

1. Human Genome Project : Projet de recherche à l'échelle mondiale qui a pour but d'identifier l'ensemble des gènes présents chez l'homme. Voir *http* : *//web.ornl.gov/sci/techresources/Human_Genome/*

Vers une modélisation de la régulation génétique

Les êtres vivants sont divisés en deux grandes familles : *les procaryotes*, organismes unicellulaires sans noyau et *les eucaryotes*, organismes dont les cellules ont un noyau qui renferme l'information génétique. Dans les deux cas, trois types de molécules portent ou permettent d'exprimer les informations génétiques d'une cellule à savoir l'acide désoxyribonucléique (ADN), l'acide ribonucléique (ARN) et la protéine (voir Annexe A pour plus de détails). Schématiquement, l'ADN est la molécule qui porte l'information génétique, l'ARN et les protéines sont les acteurs qui vont permettre à cette information de s'exprimer pour faire fonctionner le processus cellulaire. L'information génétique s'exprime au travers de deux mécanismes essentiels, la transcription (ADN — ARN) et la traduction (ARN — protéine).

De nombreux mécanismes de régulation de l'expression des gènes ont été identifiés [BO04, HGL+04, LRR+02]. La régulation se fait pour la plupart des gènes principalement au niveau de la transcription, mais elle peut également avoir lieu durant l'épissage ou le transport de l'ARN messager (chez les eucaryotes), durant leur traduction, ou lors de la maturation des protéines. Des protéines régulatrices, appelées *facteurs de transcription*, jouent un rôle important dans la transcription. Elles peuvent se lier à des sites relativement spécifiques, appelés *sites de fixation*, dans les régions promotrices des gènes et former des complexes les uns avec les autres.

Un facteur de transcription se distingue en fonction de son action sur la transcription du gène. Il est *activateur* s'il augmente son niveau d'expression (effet positif), et il est *inhibiteur*, s'il le diminue (effet négatif). Par ailleurs, un facteur de transcription peut être dual s'il est capable d'activer ou de réprimer un gène en présence d'autres facteurs de transcription. Enfin, les régulateurs peuvent eux-mêmes être régulés. Dans ce cas, ils participent à une *voie de régulation génétique*. Généralement, un gène cible est régulé par une combinaison de facteurs de transcription, et un facteur de transcription peut réguler plusieurs gènes cibles.

Réseaux de régulation génétique

Le niveau d'expression des gènes se contrôle et s'ajuste en permanence pour s'adapter au contenu cellulaire et aux conditions extérieures. Ce contrôle de l'expression des gènes est effectué grâce à un réseau de régulation génétique. Ce dernier est une collection de gènes qui interagissent les uns avec les autres via leurs produits d'expression (les protéines), permettant un contrôle mutuel de leurs taux d'expression. En d'autres termes, un composé A se déplace de manière aléatoire et a donc une certaine probabilité de rencontrer un composé B à un temps t. Si la rencontre entre les composés A et B induit un changement du niveau d'expression alors il existe une interaction entre A et B.

Concrètement, un réseau de régulation génétique peut être représenté comme un graphe dont les nœuds sont des gènes (facteurs de transcription ou gènes cibles) et les arcs sont des interactions physiques qui représentent des effets transcriptionnels. Les relations de régulation peuvent être orientées ou non. Un arc (non orienté) entre deux gènes indique seulement qu'il existe une interaction entre ces deux gènes, tandis qu'une arête (orientée) signifie que le gène source régule l'expression du gène cible en l'activant ou en l'inhibant.

Les réseaux de régulation sont des modèles abstraits permettant de capturer le maximum d'informations sur les réseaux cellulaires complets. C'est une propriété d'autant plus importante qu'il s'agit d'acquérir de l'information sur des processus de régulation qui ne sont pas mesurés physiquement. Ainsi, depuis l'avènement de la (post-)génomique, une classe de techniques à haut débit a émergé pour mesurer l'expression des gènes qui résultent des processus d'interactions génétiques : les techniques à haut débit. L'outil le plus répandu parmi ces techniques est la puce à ADN [SSDB95] (voir Annexe A). Elle permet d'estimer les niveaux d'expression de plusieurs milliers de gènes simultanément et offre ainsi la possibilité d'étudier des génomes entiers, comme par exemple le génome humain qui compte environ 30 000 gènes.

Grâce à ces techniques à haut débit, il est possible de tenter d'étudier les réseaux de régulation génétique [SK99]. L'hypothèse de base est que les niveaux d'expression de gènes fournissent des informations sur leur niveau d'activité de régulation.

Apprentissage de réseaux de régulation génétique

Nous abordons dans cette thèse le défi de l'apprentissage automatique de réseaux de régulation génétique sur la base des données d'expressions. Il s'agit d'inférer les mécanismes de régulation sous-jacents : prédire, pour chaque gène cible et dans un contexte cellulaire donné, les facteurs de transcription régulant son expression. Néanmoins, plusieurs contraintes compliquent l'apprentissage des régulations génétiques (*i.e.*, bruit, dimension, combinatoire).

En général, les données d'expression sont représentées par une matrice où les lignes représentent les gènes (gènes cibles et facteurs de transcription), et les colonnes représentent les échantillons. De nombreux problèmes sont à considérer pour l'inférence de structures à partir des données.

Le bruit

Les données d'expression dans lesquelles nous essayons de découvrir des régularités sous-jacentes à l'aide de techniques d'apprentissage artificiel ne sont pas parfaites [BN06] comme toutes les données réelles issues de processus expérimentaux. Non seulement les données peuvent comporter des valeurs manquantes, être imprécises, mais elles sont parfois non homogènes, c'est à dire qu'elles sont issues de plusieurs sources. Leur grande variabilité demande une prise en considération des aléas aussi bien expérimentaux que biologiques. Par ailleurs, et afin de diminuer le temps de calcul (*i.e.*, simplifier le problème d'inférence), certains algorithmes d'apprentissage s'appliquent sur des données catégorielles (*e.g.*, booléennes), d'où leur transformation en variables à valeurs discrètes. Cependant, cette discrétisation est aussi une source de bruit, car la transformation de données engendre une erreur entre le signal quantifié et le signal source.

Le traitement du bruit dans les données est un problème difficile. En effet, certains motifs présents dans les données d'apprentissages sont soit dûs à des erreurs (fluctuations aléatoires) soit représentent des observations authentiques (intrinsèques). Il n'est pas facile de distinguer ces deux cas, notamment dans le cas de petits échantillons.

La dimension

Les techniques à haut débit donnent souvent lieu à l'évaluation d'un grand nombre de gènes (*e.g.*, niveau d'expression de gènes dans les études de puce ADN) avec un nombre limité d'observations [MHADI06]. Ceci est communément connu sous le nom de *la malédiction de la dimension* [Dra03, RSB03]. En effet, la quantité réduite de données disponibles pour l'apprentissage d'un modèle rend le paramètrage du modèle d'apprentissage difficile. Trouver les gènes les plus pertinents [YSL07, HK08] en faisant le meilleur usage d'échantillons de données limités [BND04] est donc l'élément clé dans le problème d'apprentissage à partir de données génétiques.

La combinatoire

Les réseaux de régulations sont des systèmes complexes. En effet, chez les organismes eucaryotes pluricellulaires, qui sont dotés de régions non codantes beaucoup plus étendues, nombreuses études [LT03, Car98, FW00, HGL+04, LJFJ06] montrent que plusieurs facteurs de transcription peuvent être impliqués dans la régulation d'un gène cible. Ces régulateurs fonctionnent souvent en mode de coopération – concurrence. La complexité résulte du nombre d'interactions coopératives possibles, qui est exponentiel en fonction du nombre de régulateurs.

Algorithmes d'inférence

De nombreux algorithmes d'apprentissage de structures de régulation à partir de données d'expression ont été développés. Ces algorithmes peuvent être classés selon deux grandes familles : individuelle et ensembliste. La première est dite *individuelle* car elle infère un *seul* modèle (de structure). Un tel modèle peut être décrit — en bio-informatique — par plusieurs formalismes comme par exemple des réseaux bayésiens [FLNP00, SGS+00], des réseaux de co-expression [BK00, DLS00, dlFBHM04, BMS+05], des équations différentielles [HS96], des régressions linéaires [GHL05, MGT09, GH10], des réseaux booléens [AMK99, LFS98] ou encore à travers des arbres de décision [SKB+03]. Bien que ces modèles essaient d'être fidèles au système biologique étudié en incorporant de nombreux détails du processus de régulation, leurs succès sont partiels. En effet, l'obstacle majeur face à une telle approche est la faible quantité de données d'apprentissage. De plus, comme ces données sont bruitées, l'inférence des paramètres du modèle d'apprentissage est généralement peu précise. L'estimation empirique peut avoir une variance élevée pour différentes réalisations des données d'expression. Par conséquent, cette estimation empirique peut différer significativement de la valeur réelle, et le modèle d'inférence risque d'avoir de mauvaises performances.

Pour surmonter ces problèmes, une deuxième famille dite *ensembliste* a émergé, donnant lieu à des modèles plus fiables [HTIWG10, DMSES12, SA12, HMVLV12]. Cet axe d'apprentissage, basé sur les méthodes d'ensemble [Die97, HS90, Kun07], construit une collection de modèles d'inférence de structures *individuelles* (*e.g.*, arbres de décision, régressions linéaires, etc) dont les réponses individuelles sont, par la suite, combinées afin d'obtenir des structures (*i.e.*, agrégats) plus performantes (*e.g.*, forêts

aléatoires). Cette approche tire profit du modèle ensembliste dont plusieurs études [Bre96a, Die00b, Fri01, Web00] ont démontré que les performances sont souvent meilleures que celles d'un classifieur individuel.

La régulation de la transcription des gènes chez les eucaryotes est complexe car elle est intrinsèquement de nature combinatoire [LT03, Car98, FW00, KRK+95, HCG+00]. La coopération entre facteurs de transcription est un élément clé de ce contrôle combinatoire. Ainsi, pour comprendre et modéliser les mécanismes biologiques d'une cellule il faut nécessairement prendre en considération cette coopération dans les modèles d'inférence de structures. Néanmoins, la plupart des approches, *individuelles* ou *ensemblistes*, ne traitent pas explicitement la nature combinatoire de la transcription et simplifient le modèle local de régulation en considérant uniquement des interactions binaires (*i.e.*, une interaction entre deux entités uniquement) ignorant ainsi la nature coopérative des réseaux de régulation.

Contributions

Afin de modéliser le plus fidèlement possible les réseaux de régulations biologiques, nous proposons dans cette thèse une nouvelle démarche d'inférence de réseaux de régulation. Celle-ci répond à trois critères fondamentaux :

- Premièrement, notre approche respecte le modèle biologique (*i.e.*, contrainte de complexité) en inférant des réseaux de régulation coopératifs.
- Deuxièmement, cette approche est capable d'inférer des réseaux de régulation fiables et de garder des bonnes performances dans des conditions d'apprentissage difficiles (*i.e.*, contrainte de bruit, contrainte de dimension).

Afin de fournir une solution performante respectant ces critères, nous nous appuyons sur un algorithme développé au sein de notre équipe, appelé LICORN [ENBF+07].

LICORN est une méthode de fouille de données discrète qui décompose la tâche d'apprentissage de structures pour chaque gène cible en trois étapes indépendantes : (i) l'extraction de complexes de régulateurs fréquents, (ii) la génération de réseaux locaux candidats, composés de complexes de régulation étiquetés, et (iii) la sélection, parmi l'ensemble des réseaux candidats, du meilleur suivant une fonction de score discrète. Cette fonction de sélection est supervisée et permet de mesurer la capacité d'un ensemble de régulateurs étiquetés à estimer le niveau d'expression d'un gène cible sur la base de la mesure des *moindres écarts absolus* (MAE).

Le choix de ce composant de base est motivé principalement par deux raisons :

- LICORN permet d'inférer des réseaux locaux coopératifs étiquetés.
- LICORN *passe à l'échelle* permettant d'apprendre des réseaux de régulation complexes comme ceux de l'humain.

Bien que LICORN présente plusieurs avantages, il doit être étendu pour deux raisons : premièrement, il sélectionne *un seul* réseau candidat par gène, ce qui détériore sensiblement ses performances (*i.e.*, peu d'interactions réelles prédites). De plus, la sélection de ce réseau est basée sur un choix *aléatoire* parmi un ensemble de candidats ayant la même valeur de score MAE, ce qui rend la phase de sélection peu précise. Deuxièmement, l'absence d'un processus de classement au niveau des interactions dégrade sensiblement

les performances de l'algorithme.

Dans cette thèse, nous avons proposé plusieurs solutions afin de remédier aux limitations de LICORN, tout en exploitant ses points forts.

Tout d'abord, nous avons conçu une nouvelle démarche qui divise la tâche d'apprentissage en trois phases. Premièrement, nous avons extrait un ensemble de réseaux candidats pour chaque gène cible à l'aide de la méthode LICORN. Ensuite, nous avons élaboré une méthodologie *de sélection d'ensemble* de réseaux à partir de l'ensemble des réseaux candidats. Afin de garantir que les réseaux sélectionnés soient à la fois *précis* et *divers*, nous avons basé ce processus sur les méthodes d'ensemble [Die00a, HS90, UN96]. Dans une première version, nous avons utilisé un algorithme glouton de sélection qui n'ajoute un réseau à l'ensemble sélectionné que s'il améliore les performances de l'ensemble courant.

Enfin, nous proposons un *processus original de classement*, qui utilise une technique numérique : la régression linéaire. Ce dernier répond a deux objectifs : le respect de la structure des réseaux locaux (*i.e.*, complexe de régulation) et la quantification de la fiabilité de chaque interaction.

Cette démarche est implémentée dans un outil appelé SELECTNET [CEN+] et évaluée sur les données *In silico* du challenge DREAM5 [2]. Les résultats obtenus permettent de répondre aux limites de LICORN, mais soulèvent une question importante concernant le rôle de la diversité dans le processus de sélection. En effet, à travers nos expériences, nous avons constaté une dégradation de performances suite à la sélection de réseaux divers. Ceci est contradictoire avec la théorie des méthodes d'ensembles qui stipule qu'un ensemble doit être divers et précis à la fois afin de garantir une amélioration de la performance de classifisation.

Afin de respecter la définition de l'approche ensembliste, nous avons proposé une extension de SELECTNET, appelée SETNET [CERS13], dans laquelle nous introduisons le bagging [Bre96a] pour produire de la diversité dans les réseaux candidats. Cette technique consiste à perturber les données d'apprentissage avant l'inférence, ce qui conduit à des changements dans les modèles appris, permettant ainsi la découverte de nouveaux réseaux candidats.

Sur les données DREAM5, les résultats montrent une forte amélioration de performances. Les réseaux inférés par SETNET sont largement plus riches en interactions réelles que ceux inférés par SELECTNET. Cependant, l'analyse de ces réseaux révèle qu'un grand nombre d'interactions non réelles sont aussi sélectionnées. Ce constat met en doute la phase de sélection, et en particulier la capacité de la fonction de score discrète à discriminer les bonnes interactions des mauvaises, générant ainsi un choix de réseaux inadéquat.

Pour traiter cette problématique, nous avons substitué l'approche discrète de sélection par une nouvelle approche numérique. Ainsi, nous avons proposé une méthode *hybride* appelée H_SETNET [CNS+13, CNS+14] qui génère des réseaux candidats suivant un modèle local *discret* de régulation, puis sélectionne les plus précis selon une méthode de sélection *numérique*.

H_SETNET est évalué sur les données DREAM5 et est comparé à deux algorithmes de l'état de l'art : ARACNE [MNB+06] et GENIE3 [HTIWG10] (*i.e.*, la méthode gagnante du challenge DREAM5). Deux enseignements peuvent être tirés des résultats :

2. Voir http ://wiki.c2b2.columbia.edu/dream/index.php ?title=D5c4

– H_SetNet est capable de discriminer les bonnes interactions, diminuant ainsi le nombre d'interactions artefacts inférées (elle est la méthode qui infère le moins de mauvaises interactions, comparée à Aracne et Genie3).
– H_SetNet garde de bonnes performances face à la contrainte de sous échantillonnage (elle est la méthode la moins sensible aux échantillons de faible taille par rapport à Aracne et Genie3).

Dans le but de confirmer ces bonnes performances, nous avons évalué la capacité de H_SetNet à identifier des relations de régulations du cancer. En collaboration avec Mohamed Elati et Rémy Nicole de l'équipe MEGA de l'institut de Biologie Systémique et Synthétique (iSSB), nous avons réalisé une étude expérimentale de H_SetNet sur les données *du cancer de la vessie* de l'institut Curie (groupe oncologie moléculaire de François Radvanyi). Contrairement à l'évaluation des méthodes d'inférence sur Dream5, les relations de régulations réelles pour les données humaines ne sont pas connues. Pour cette raison, nous avons dû utiliser plusieurs types de données auxiliaires (*i.e.*, TFBS, ChIP et STRING) afin d'évaluer la pertinence de nos prédictions.

Ces résultats montrent que H_SetNet est la méthode inférant le plus grand nombre d'interactions existantes confirmées sur les données TFBS et ChIP, en comparaison à Genie3 et à Aracne. En plus, une partie significative des relations coopératives entre régulateurs prédites par H_SetNet existe dans la base des interactions protéiques STRING.

Le manuscrit est organisé de la façon suivante :
– Le *Chapitre 1* présente l'environnement méthodologique de l'apprentissage ainsi que la modélisation de réseaux de régulation à partir de données de transcriptome. Dans ce chapitre, nous dressons un état de l'art de la première famille *individuelle* d'inférence de réseaux de régulation.
– Le *Chapitre 2* est consacré à la théorie des méthodes d'ensemble. Nous détaillons les approches de la famille *ensembliste* pour l'inférence de réseaux de régulation.
– Le *Chapitre 3* est une étude critique de Licorn. La formalisation du modèle de régulation coopérative et l'algorithme d'apprentissage y sont présentés en détail. Une étude expérimentale des réseaux de régulations inférés sur des données du challenge Dream5 est faite afin de cerner ces limites. Enfin, des pistes d'extension sont présentées.
– Le *Chapitre 4* présente nos algorithmes SelectNet et SetNet. Nous abordons la construction d'*ensemble* de réseaux de régulation sur la base d'un modèle *discret*. Les algorithmes d'apprentissage proposés y sont détaillés. Une étude expérimentale est réalisée sur les données Dream5.
– Le *Chapitre 5* décrit notre algorithme H_SetNet. Dans ce cadre, une méthode *numérique* alternative à la méthode discrète pour la sélection d'ensembles de réseaux implantée dans SetNet est proposée. Une étude expérimentale sur des données Dream5 et des données humaines de cancer de la vessie est accomplie.
– Enfin, la dernière partie est consacrée à la conclusion et à la discussion de plusieurs directions de recherche prolongeant ce travail de thèse.

Première partie

Étude bibliographique

Chapitre 1

Inférence des réseaux de régulation

Sommaire

Ce chapitre est consacré à la présentation des méthodes individuelles d'inférence de réseaux de régulation à partir des données d'expressions de gènes. Nous introduisons l'apprentissage de structure pour les réseaux de régulation et nous dressons ensuite une synthèse des travaux effectués en nous focalisant sur les méthodes *statiques* (*i.e.*, qui ignorent les aspects temporels). Pour ce faire, nous distinguons deux axes d'approches d'inférence de réseaux de régulations, celles qui apprennent les structures de régulation à partir des données discrètes et celles qui les apprennent à partir des données continues.

1.1 Apprentissage automatique de structures

L'apprentissage automatique consiste en la conception de programmes qui s'améliorent avec l'expérience. Les aspects les plus importants à prendre en compte pour un système d'apprentissage automatique sont ses capacités de généralisation [Mit82], d'optimisation [Vap98] et d'intelligibilité [Mic83].

Les méthodes d'apprentissage, issues de l'intelligence artificielle et des statistiques exploratoires, sont très variées et peuvent être classées de diverses manières, selon qu'elles soient supervisées ou non supervisées, intégratives ou non intégratives, déterministes ou probabilistes [GS02, CM02, RDS10]. En effet, l'apprentissage automatique englobe toute méthode permettant de construire automatiquement un modèle à partir de données. Ces données peuvent prendre différentes formes, quantitatives ou qualitatives, logiques ou ordinales, continues ou discrètes, à faible ou à grande dimension.

Au même moment, le succès rencontré par les techniques de puces à ADN pour mesurer l'expression des gènes à grande échelle a conduit à l'émergence d'algorithmes d'inférence des réseaux de gènes à partir de ces données d'expression. L'objectif est de proposer, à partir de données d'expression, des interactions candidates entre les gènes

qui pourront être ensuite validées par des expérimentations biologiques. L'apprentissage automatique, reconnu en 2003 parmi les dix technologies émergentes [MIT03], offre pour cette problématique un cadre à la fois théorique, méthodologique et enfin économique.

Le problème d'inférence des réseaux de régulation a été étudié depuis de nombreuses années dans la littérature et de nombreux algorithmes ont été proposés. Smet et Marchal [RDS10] ont établi une catégorisation de ces méthodes (voir la Figure 1.1). Tout d'abord, ils distinguent les méthodes *supervisées* des *non supervisées*. Les méthodes supervisées exploitent des connaissances préalables du réseau pour guider l'inférence de réseau, alors que les méthodes non supervisées n'ont aucune connaissance préalable. Par ailleurs, l'utilisation d'informations auxiliaires permet de raffiner encore plus cette classification en définissant deux autres familles à savoir les méthodes *intégratives* et *non intégratives*. Les méthodes non intégratives utilisent uniquement des données d'expression pour l'inférence et ont donc pas recours à des informations auxiliaires. En revanche, les méthodes intégratives exploitent d'autres types d'information en plus des données d'expression, comme par exemple l'information relative aux motifs de séquences qui servent de sites de fixation pour des facteurs de transcription.

Finalement, les méthodes dites *directes* ne considèrent que les interactions par paires tandis que les méthodes *par module* recherchent des ensembles de gènes régulés par les mêmes facteurs de transcription.

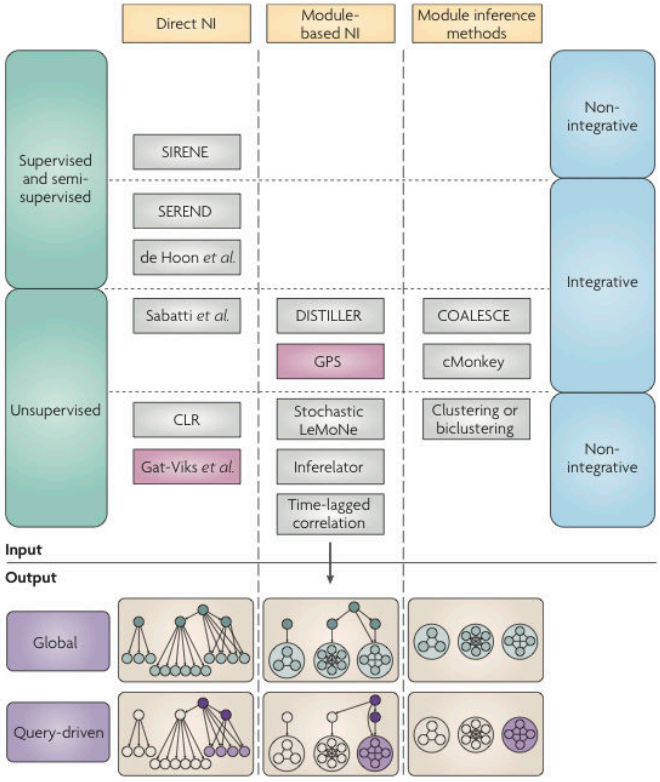


FIGURE 1.1 – Catégorisation des méthodes d'inférences de réseaux de régulation (Figure prise de [RDS10])

Dans le cadre de ce chapitre, nous proposons une classification basée sur le type de

données traitées : *continues* ou *discrètes*, ce qui nous permet de distinguer deux catégories de méthodes *statiques* d'inférence de réseaux de régulation. Ainsi nous parlerons, dans ce qui suit, d'approches continues et d'approches discrètes.

1.2 Approches discrètes d'inférence de réseaux

Les approches discrètes semblent assez bien adaptées au systèmes biologiques. Dans l'idéalisation booléenne des réseaux génétiques, on modélise un gène par une variable booléenne $\{1, 0\}$. Un gène est donc soit transcrit (1), soit non transcrit (0).

Les influences positives ou négatives d'un gène sur les autres peuvent aussi être modélisées par des fonctions booléennes. En d'autres termes, suite à une action de régulation d'un facteur de transcription, un gène donné peut changer d'état rendant ainsi son niveau d'expression sur-exprimé (*i.e.*, activé). Il peut aussi être réprimé, rendant son niveau d'expression sous-exprimé (*i.e.*, inhibé). Dans ce cas, la modélisation est en trois états $\{1, 0, -1\}$ *i.e.*, sur-exprimé (1), neutre (0), ou sous-exprimé (-1).

Les méthodes discrètes, bien qu'éloignées des modèles biologiques réels, offrent divers applications dans le cadre de la régulation génétique, notamment pour la modélisation de systèmes booléens dynamiques [Kau69a, AKMM98, LSYH03]. Nous présentons, dans ce qui suit, les approches statiques discrètes les plus représentatives dans l'inférence de réseaux de régulation.

1.2.1 Les réseaux bayésiens

Les premiers travaux sur les réseaux bayésiens ont été menés par J. Pearl [Pea78]. Ces modèles permettent de décrire les relations de probabilités conditionnelles entre des faits. Cette représentation repose sur un graphe orienté sans cycle (Directed Acyclic Graph) dans lequel chaque noeud, c'est-à-dire chaque variable du modèle, possède une table de probabilités conditionnelles, et où chaque arc représente une dépendance directe entre les variables reliées. Ces réseaux représentent alors la distribution de probabilités jointes à l'ensemble des variables de manière compacte, en s'appuyant sur les relations d'indépendance conditionnelle.

Un réseau bayésien permet de prendre en compte des distributions de probabilité conditionnelle paramétrique ou non paramétrique et des variables discrètes ou continues. Nous les classons ici dans les approches discrètes vu qu'un nombre important de travaux [FLNP00, SGS$^+$00] les a appliqués sur des données discrètes. Les réseaux bayésiens sont sans doute les modèles les plus utilisés pour l'inférence de réseaux de régulation. Les premières approches utilisant ce type de modèle pour l'apprentissage de réseaux de régulation à partir de données d'expression ont été initiées par Friedman et al [FLNP00] et Spirtes et al. [SGS$^+$00].

Dans ces travaux, chaque gène est modélisé par une variable discrète et le réseau de régulation est représenté par un graphe orienté acyclique qui suggère l'influence causale entre les gènes. Les sommets sont des gènes et les arcs codent les dépendance. La Figure 1.2 montre un exemple d'un tel réseau. Les variables sont associées aux noeuds du réseau et l'absence de lien indique une indépendance entre les noeuds.

L'apprentissage de réseaux bayésiens est un champ de recherche très actif et consiste

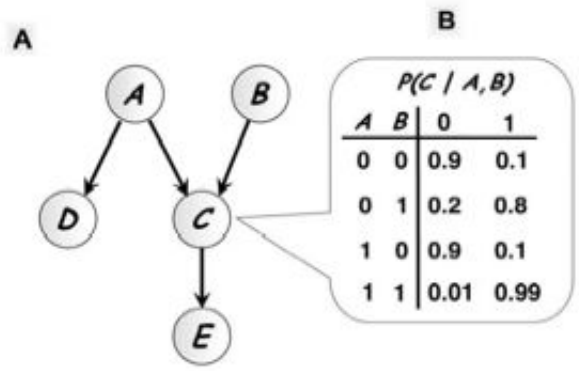


FIGURE 1.2 – (A) Un réseau bayésien constitué de 5 variables aléatoires. (B) Un exemple de la distribution conditionnelle sous la forme du produit qui spécifie $P(C|A, B)$

à trouver un réseau bayésien modélisant les données disponibles. Il existe deux grandes familles d'approche pour apprendre la structure d'un réseau bayésien à partir de données.

La première approche fondée sur les contraintes consiste à rechercher les différentes relations causales qui existent entre les variables et à tester les indépendances conditionnelles. L'idée est de chercher une structure de réseau cohérente avec les dépendances et indépendances observées. Cette approche traite un nombre très limité de variables et est moins utilisée que la deuxième pour l'inférence de réseaux de régulation.

Quant à la deuxième approche, elle mesure l'adéquation des (in)dépendances codées dans le réseau avec les données, en associant un score, le but étant de chercher la structure qui donnera le meilleur score dans l'espace des graphes acycliques. Cette approche, la plus utilisée pour l'inférence de réseaux de régulation, est bien fondée, c'est-à-dire qu'avec suffisamment de données, l'apprentissage converge vers une structure performante. Cependant, la recherche d'une structure optimale est un problème NP-difficile [Pea78, CHM04]. La recherche exhaustive du meilleur réseau guidée par une fonction de score, est d'un point de vue informatique impossible. En effet, ce qui rend cette recherche irréalisable est le nombre super-exponentiel en fonction du nombre de régulateurs des graphes candidats. Pour résoudre ce problème, un certain nombre d'heuristiques de recherche dans l'espace des graphes ont été proposées [Fri04, Chi02], comme par exemple, restreindre l'espace d'hypothèses en limitant le nombre de parents possibles pour chaque noeud ou encore effectuer une recherche gloutonne dans les réseaux candidats. En effet, Pe'er et al. proposent dans leurs travaux [PRT02, PTR06] de mette en avant les connaissances biologiques pour limiter l'ensemble des parents candidats aux protéines régulatrices (*i.e.*, facteurs de transcription). De plus, ils imposent une contrainte sur la topologie locale du réseau, en bornant le nombre maximal de régulateurs d'un gène cible (*e.g.*, trois régulateurs).

Ces heuristiques ont été appliquées dans l'algorithme MINREG [PRT02]. Ce dernier recherche une approximation de l'ensemble minimal de régulateurs actifs AR pour un ensemble de gènes cibles. En effet, il est guidé par un score local, basé sur l'information mutuelle, qui évalue le degré de dépendance entre un sous-ensemble de régulateurs et un gène cible. Il calcule itérativement et de manière gloutonne l'ensemble minimal AR qui contrôle un ensemble de gènes. A chaque étape, il ajoute à AR le régulateur r qui permet de maximiser le score global de corrélation entre chacun des gènes cibles et tous

les sous-ensembles de $AR \cup r$ de taille $\leq d$. L'hypothèse de cette méthode est que le réseau global contient un petit nombre de régulateurs ayant de nombreuses cibles (voir Figure 1.3).

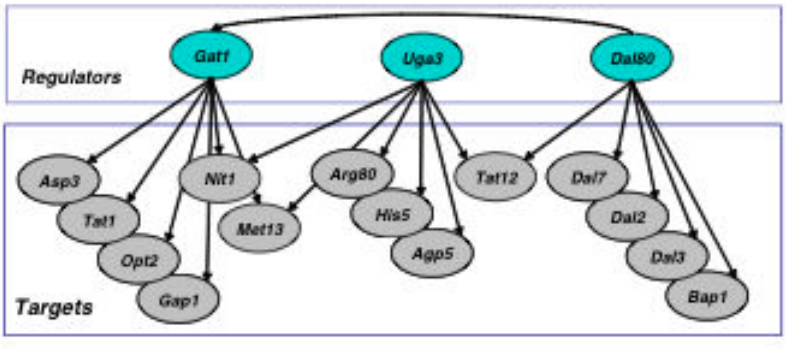


FIGURE 1.3 – Réseau de régulation extrait de [PTR06]. Le niveau supérieur représente les régulateurs et le niveau inférieur contient les gènes cibles.

Les réseaux bayésiens sont très utilisés dans l'inférence de réseaux de régulation mais souffrent de divers inconvénients qui sont majoritairement liés à la complexité des algorithmes. Les problèmes sont pratiquement tous de complexité non polynomiale, conduisent à développer des algorithmes approchés (approximatifs), dont le résultat n'est pas garanti pour des problème de grande taille (nombreux régulateurs/gènes cibles et peu d'échantillons).

1.2.2 Les réseaux de co-expression

Plusieurs travaux dans la littérature [BK00, DLS00, dlFBHM04, BMS+05] construisent des réseaux d'interactions entre paires de gènes. Ces méthodes diffèrent essentiellement par la mesure utilisée pour décider d'une interaction entre deux gènes. Les réseaux les plus étudiés sont les réseaux de co-expression de gènes, dans lesquels deux gènes sont reliés si et seulement s'ils sont co-exprimés.

Par exemple, Butte et Kohane [BK00] ont développé une méthode de construction de réseaux de co-expression basée sur l'information mutuelle (*MI*), appelée *relevance networks*. Pour ce faire, ils calculent la MI entre toutes paires de gènes, puis, à l'aide d'un test de permutation, ils considèrent que deux gènes sont co-exprimés si et seulement si leur information mutuelle est supérieure à l'information mutuelle maximale obtenue pour ces deux gènes dans les données permutées.

Diverses améliorations ont été proposées pour tenter de distinguer les interactions directes des indirectes dans les *relevance networks*. Les plus importantes sont celles des algorithmes ARACNE [MNB+06] et CLR [FHT+07].

ARACNE filtre les interactions indirectes des triplets de gènes avec l'inégalité de traitement des données (Data Processing Inequality DPI). En effet, il affirme que si les gènes (g_i, g_j) et (g_j, g_k) sont directement en interaction, et (g_i, g_k) interagit indirectement par j, alors $MI(g_i, g_k) \leq min(MI(g_i, g_j), MI(g_j, g_k))$. La Figure 1.4 illustre un exemple de DPI de quatres gènes $g1$, $g2$, $g3$, et $g4$. Bien que toutes les paires de gènes aient

probablement un lien après l'estimation de l'information mutuelle, le DPI élimine des liens et déduit le chemin le plus probable. En effet : $g_1 \leftrightarrow g_3$ sera éliminé parce que $MI(g_1, g_2) > MI(g_1, g_3)$ et $MI(g_2, g_3) > MI(g_1, g_3)$. $g_2 \leftrightarrow g_4$ sera éliminé parce que $MI(g_2, g_3) > MI(g_2, g_4)$ et $MI(g_3, g_4) > MI(g_2, g_4)$. $g_1 \leftrightarrow g_4$ sera éliminé de deux façons : d'abord, parce que $MI(g_1, g_2) > MI(g_1, g_4)$ et $MI(g_2, g_4) > MI(g_1, g_4)$, et ensuite parce que $MI(g_1, g_3) > MI(g_1, g_4)$ et $MI(g_3, g_4) > MI(g_1, g_4)$. Enfin, g_1, g_2, g_3, et g_4 sont connectés dans une relation linéaire.

Cette étape d'élimination réduit le nombre de faux positifs, ce qui augmente fortement la précision de l'algorithme mais en revanche, diminue sa sensibilité en éliminant de nombreuses paires à faibles interactions. Dans [MNB+06], la méthode a pu recouvrir des interactions génétiques dans les cellules de mammifères et donner de meilleurs résultats sur plusieurs tâches d'inférence en se comparant avec les *relevance networks*.

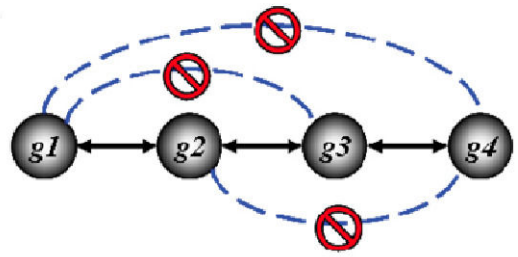


FIGURE 1.4 – Un exemple de l'inégalité de traitement des données (DPI) de quatre gènes $g1$, $g2$, $g3$, et $g4$. La figure est extraite de [MNB+06].

Quant à l'algorithme CLR, une correction d'estimation du score sur la base de distribution empirique est faite sur tous les scores MI. Pour chaque gène, le seuil de MI est calculé dans son propre réseau. Ensuite, pour chaque interaction facteur de transcription – gène cible, le score MI est comparé au seuil du facteur de transcription ainsi qu'au gène cible et est transformé en un *z-score*. Enfin, un nouveau score *z-score* est estimé. Ce dernier dépend de la distribution des scores MI pour tous les régulateurs possibles d'un gène cible et de la distribution des scores MI pour toutes les cibles possibles d'un régulateur. Les interactions sont ensuite classées par ordre décroissant de *z-score*.

Dans le même contexte des réseaux de co-expression, nous trouvons d'autres heuristiques de simplification de graphes de MI, citons entre autres l'algorithme *MRNET* [MKLB07] et la méthode *C3NET* [AES10, AES11]. *MRNET* formule le problème de l'inférence comme une série de procédures de sélection supervisée adoptant le principe de pertinence maximale–redondance minimale (*mRMR*) [DP05] qui est une technique supervisée efficace pour la sélection de variable. L'idée de ce principe consiste à sélectionner parmi les variables les moins redondants (*i.e.*, redondance minimale) ceux qui ont l'information mutuelle la plus élevée avec la cible (*i.e.*, pertinence maximale). *MRNET* étend ce principe de sélection de variable à des réseaux de régulation afin de déduire les relations gène–gène à partir de données d'expression.

D'une autre manière, *C3NET* filtre les arêtes jugées inutiles dans les réseaux appris et ce en ne conservant entre deux gènes que les arêtes ayant au moins pour l'un des deux une estimation MI maximale.

Ces réseaux de *co-expression* présentent des inconvénients majeurs. En effet, afin de décider d'ajouter ou non une interaction entre deux gènes, le coût des tests statistiques pour chaque paire de gènes peut être assez élevé surtout si on considère tous les gènes du génome. En plus, toutes les méthodes citées jusque là sont des méthodes globales cherchant à inférer des grands réseaux de régulation. Ces dernières cherchent à identifier des similarités entre paires de gènes et sont fondamentalement limitées puisque seules des interactions entre paires de gènes sont considérées, alors que plusieurs études [HGL+04, LJFJ06] montrent que plusieurs facteurs de transcription différents peuvent être impliqués dans la régulation d'un gène (facilement une dizaine). Ces régulateurs fonctionnent souvent en mode de coopération–concurrence, ce qui explique les limites des approches globales proposées, qui simplifient le modèle de régulation local.

1.2.3 Les réseaux locaux coopératifs

Elati et al. [ENBF+07] ont proposé une démarche locale alternative aux approches globales appelée LICORN. Cette méthode cherche un réseau de régulation local complexe pour chaque gène étudié. En collaboration avec des biologistes, ils ont proposé un modèle local logique de régulation qui couvre les modes de coopération–concurrence opérant lorsque plusieurs régulateurs agissent sur un même gène cible [NKS05, CWC06]. Une phase de distinction selon leur mode de régulation (activateurs ou inhibiteurs) est mise en place.

LICORN décompose la tâche d'apprentissage de structures en trois étapes indépendantes : une première étape, fondée sur une technique existante de fouille de données [AIS93], calcule les sous-ensembles de co-régulateurs fréquents. Puis, dans une deuxième étape, il recherche efficacement pour chaque gène cible l'ensemble de ses co-régulateurs candidats. Ces ensembles sont de co-régulateurs fréquents précédemment calculés qui varient conjointement avec le gène cible dans un nombre significatif d'échantillons. Cet ensemble de co-régulateurs candidats, de taille modeste, est ensuite affiné grâce à une recherche guidée des couples (co-activateurs, co-inhibiteurs) formant les réseaux candidats, et ce, suivant une fonction de score local. Dans la dernière étape, LICORN sélectionne, pour chaque gène cible, le réseau de régulation local de meilleur score. L'algorithme a été appliqué sur deux jeux de données réels de la levure. Il a permis de découvrir des relations combinatoires entre régulateurs et leurs gènes cibles compatibles avec des résultats expérimentaux publiés. Les régulations coopératives inférées par LICORN n'ont pas été identifiées par d'autres approches fondées sur des modèles bayésiens ou arbre de décision [PRT02, SSR+03, MKW+04].

1.3 Approches continues d'inférence de réseaux

Les méthodes discrètes présentées dans la section précédente peuvent révéler des propriétés de réseaux importantes, néanmoins, elles sont limitées lorsqu'il s'agit de saisir certains aspects importants de la dynamique des réseaux. En effet, ces algorithmes d'inférence sont essentiellement fondées sur l'algèbre de tables de probabilités et modélisent des distributions de probabilités discrètes. Par contre les méthodes continues modélisent

explicitement les changements du niveau d'expression des gènes permetrant ainsi une description plus détaillée–précise des relations de régulation.

Nous exposons, dans cette section, divers méthodes continues qui ont confirmé leur performance dans l'identification d'interactions génétiques.

1.3.1 Les méthodes à noyaux

La logique derrière les méthodes à noyau [SS02, SBE99] est que, même si aucune hypothèse n'est faite concernant la relation entre le niveau d'expression mesuré d'un facteur de transcription et ces gènes cibles, si deux gènes sont régulés par le même facteur de transcription, alors ils sont susceptibles de présenter des profils d'expression similaires. Ces méthodes sont généralement précises et ont l'avantage de traiter tous types de données pour lesquelles une mesure de similarité (un noyau) peut être définie.

Qian et al. [QLYs03] présentent une approche à base de séparateurs à vastes marges (en anglais *Support Vector Machines* (*SVM*)) pour prédire les gènes cibles d'un facteur de transcription en identifiant les relations entre leurs profils d'expressions. Les *SVMs* sont une méthode d'apprentissage supervisée fondée sur des méthodes dites à noyau. La démarche consiste à la définition d'une mesure de distance dans l'espace des caractéristiques induit par le noyau, et l'application de séparateurs entre exemples positifs et négatifs. Dans ce travail, le classifieur est entraîné et testé sur un ensemble d'exemples de relations positives et négatives de couples "facteur de transcription–gène cible". Cependant, le nombre de relations de régulation positives connues est relativement limité (en particulier dans le cas humain), et cette formalisation du problème ne prend en compte que des interactions binaires.

Plus récemment, Geurts et al. [GWdAB06] ont montré la possibilité d'utiliser un noyau sur la sortie des méthodes arbre, qui leur permettent de traiter des sorties complexes. L'algorithme proposé *Output Kernel Tree* (OK3) est une sorte d'extension des méthodes d'arbre (arbres de régression, boosting d'arbres) qui considère un espace de sortie muni d'une fonction noyau. Dans ce cadre, les auteurs ont utilisé OK3 pour compléter des réseaux d'interaction protéines-protéines à partir d'une matrice d'entrée de données d'expression [GTDdB07].

Ultérieurement, Mordelet et al. [MV08] proposent la méthode *SIRENE*. Cette dernière est une méthode supervisée qui décompose le problème global de l'inférence de réseau de régulation en un grand nombre de problèmes de classification binaire locaux résolus par *SVM*. Chaque réseau local vise à identifier de nouveaux gènes cibles pour un facteur de transcription particulier. Le paradigme biologique de base utilisé par *SIRENE* suppose que si un gène A est régulé par un gène B et qu'un gène A' est similaire au gène A alors il est très probable que le gène A' est régulé par le facteur de transcription B.

1.3.2 Les méthodes d'arbres de décision

La technique d'arbre de décision [SL91, BA97] part de l'idée simple que la classification d'un objet pourrait être réalisée par une série successive de tests sur ses attributs. Le principe d'une telle technique est d'organiser l'ensemble des données comme un arbre : une feuille de cet arbre désigne une des C classes (à chaque classe peut correspondre plusieurs feuilles) et à chaque noeud interne est associé un test portant sur un ou plusieurs

attributs de l'espace de représentation. La réponse à ce test désignera le fils du noeud vers lequel on doit aller. La classification s'effectue donc en partant de la racine pour poursuivre de manière récursive le processus jusqu'à ce que l'on rencontre une feuille. Autrement dit, d'un nœud racine à un nœud feuille, une décision est prise à chaque nœud intérieur indiquant la voie à suivre. Bien que les prédictions de gènes soient binaires dans cette approche (le gène est prévu être «actif» ou «inactif»), ce modèle utilise des valeurs d'expression continues.

Soinov et al. utilisent les arbres de décision pour inférer des réseaux de régulation [SKB+03]. Leur technique permet d'identifier les gènes affectant le gène cible directement à partir du classifieur (voir exemple de la Figure 1.5).

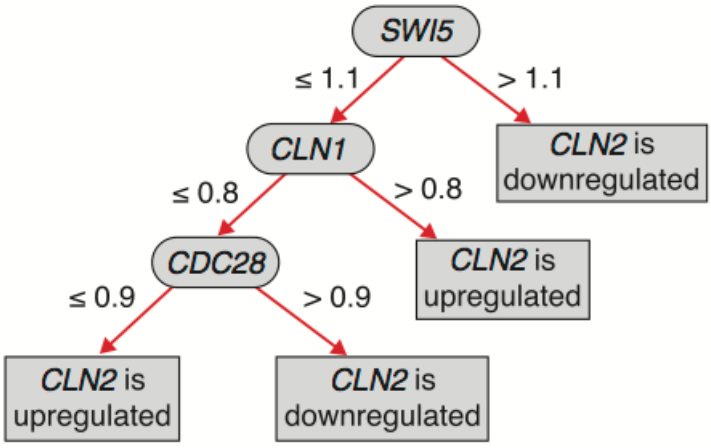


FIGURE 1.5 – Exemple d'un arbre de décision pour un gène "CLN2" de S. cerevisiae. CLN2 est le gène prédit ; SWI5, CLN1 et CDC28 sont les gènes explicatifs (régulateurs). Le seuil de l'expression des gènes, expliquant une décision, marque toutes les arètes. Figure extraite de [SKB+03]

Plus récemment, Middendorf et al. [MKW+04] proposent un protocole de classification supervisée, appelé *GeneClass*. Ce protocole est basé sur les arbres de décision, et ce en intégrant les données d'expression avec les prédictions de liaison des facteurs de transcription sur les régions promotrices de leurs gènes cibles. Ils utilisent un logiciel de découverte de sites de fixation (Match [MFG+03]) pour mettre en évidence les facteurs de transcription susceptibles d'expliquer la variation d'expression des gènes cibles. En effet, *GeneClass* prédit les classes $+1$ et -1 pour les gènes cibles, correspondant à l'état sur et sous-exprimé, à partir de ses régulateurs potentiels (présence d'un site de fixation).

1.3.3 Les méthodes de régression linéaire

De nombreuses recherches ont utilisé la régression linéaire pour inférer des réseaux de régulation. Schlitt et Brazma [SB07] proposent de générer un arbre de régression pour chaque gène à partir de tous les autres gènes. Ces arbres détectent les dépendances linéaires entre les gènes et établissent des groupes de dépendance gène – gène. Un graphe non orienté global sera ensuite construit entre tous les gènes, tout en ne gardant que les paires statiquement significatives. L'élimination des paires non-significatives est faite suivant la

méthode de Benjamini et Yekutieli [BH00]. Cette méthode statistique permet le contrôle du taux attendu des fausses prédictions parmi toutes les prédictions faites. L'approche proposée est nommée *REGNET* [SB07] (voir Figure 1.6).

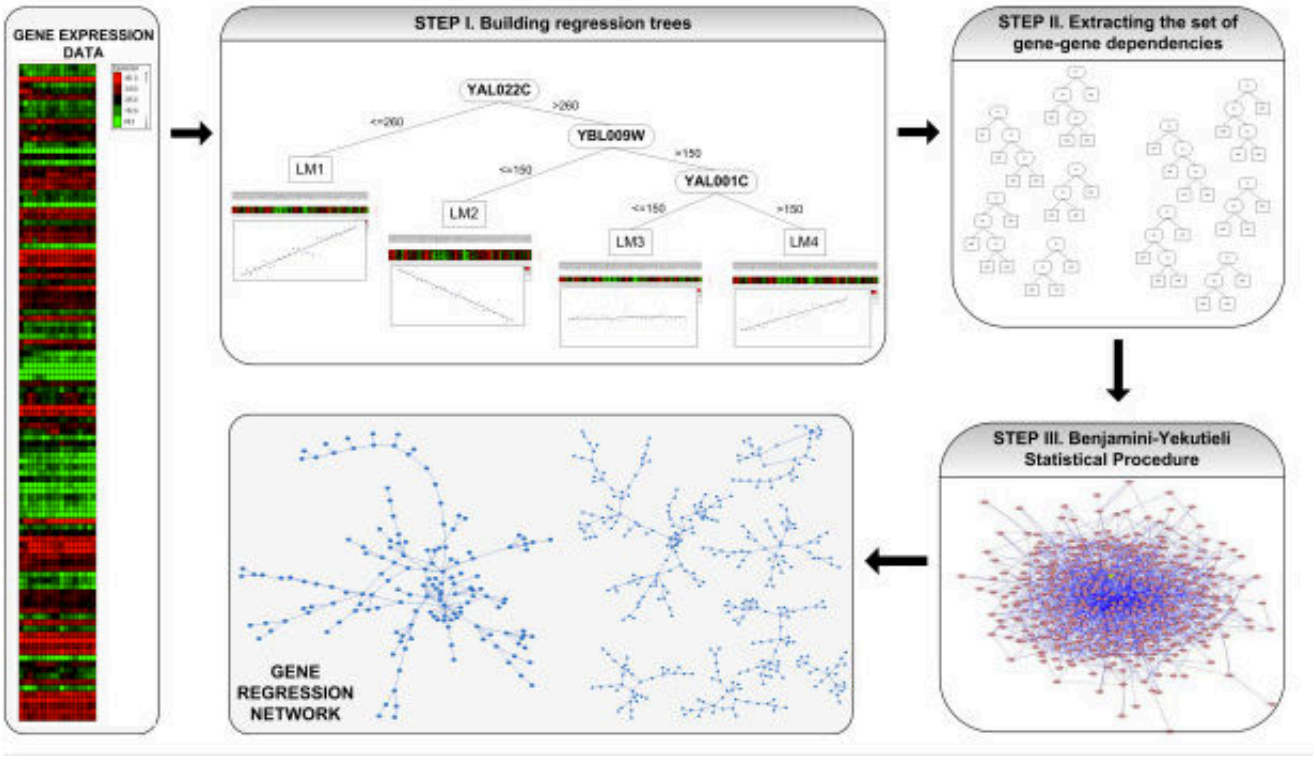


FIGURE 1.6 – Les étapes de *REGNET*. Figure extraite de [SB07]

Gustafsson et Hornquist utilisent de nouvelles formes de régression linéaire, telles que la régression LASSO [Tib94], la régression LARS [EHJT04] et la régression ELASTIC-NET [ZH05]. Leurs travaux [GHL05, MGT09, GH10] prouvent l'efficacité de telles méthodes dans l'identification et la sélection des gènes adéquats pour la régulation génétique.

Dans un premier travail [GHL05], ils élaborent un algorithme de sélection de régulateurs pour chaque gène cible en se basant sur la régression LASSO. Cette forme de régression, permet d'identifier les gènes peu importants et de les éliminer afin de ne garder que les gènes importants. Au final, l'ensemble de gènes corrélés sélectionné est largement plus petit que l'ensemble de départ. Les interactions génétiques sélectionnées ont été validées sur les interactions connues de la levure S. cerevisiae [SSZ+98]. Bien que l'utilisation d'un tel modèle linéaire soit confirmée expérimentalement, il présente une limite. En effet, en présence de variables explicatives corrélées, une seule variable est arbitrairement choisie, les autres sont écartées (*i.e.*, coefficient mis à zéro).

Ultérieurement, Gustafsson et Hornquist [MGT09] proposent un algorithme performant à base d'estimations des paramètres de la régression LARS. LARS modélise l'expression d'un gène cible comme une combinaison linéaire de l'expression de ses facteurs de transcription. A partir d'un modèle constant où aucun facteur de transcription n'est

utilisé, l'algorithme ajoute de manière gloutonne les facteurs de transcription dans le modèle afin d'affiner la prédiction du gène cible. Le résultat de la sélection de variable est statistiquement valide et s'apparente au critère Lasso [Wei05, HTFF05]. Dans la pratique, après n itérations de Lars, un classement de n facteurs de transcription sélectionnés est formé, indiquant leur capacité à prédire l'expression du gène cible en question. L'algorithme produit ensuite un graphe orienté complet, où un rang est attribué à chaque arête. Cette méthode a été appliquée sur des données *In Silico1* et *In Silico2* du challenge Dream2 [1] [SPC09] et a fourni le meilleur résultat dans le cas de l'identification d'un réseau de régulation orienté. Cependant, Lars peut être très sensible et instable s'il existe une forte corrélation entre les différentes variables explicatives.

Enfin, dans [GH10], une nouvelle forme de régression appelée Elastic-net a été appliquée. Cette forme surmonte les limites du Lasso [ZH05]. Rappelons que dans le cas d'un groupe de variables fortement corrélées, Lasso tend à sélectionner une variable et ignorer les autres en mettant leurs poids à 0 alors que Elastic-net estime l'importance des variables et les normalise sans les mettre à 0. En outre, Elastic-net est particulièrement utile lorsque le nombre de variables est beaucoup plus grand que le nombre d'observations contrairement au Lasso qui n'est pas très satisfaisant dans ce cas de figure. L'algorithme proposé a gagné le challenge de Dream3 [JMSR$^+$10]. Les clés de sa performance sont liées à l'intégration de données d'expression externes décrivant d'autres conditions expérimentales non présentes dans le challenge [MG09], ainsi qu'à la validation, où chaque donnée externe n'est utilisée que si elle augmente les performances.

1.4 Conclusion

Nous avons présenté dans ce chapitre, plusieurs méthodes d'inférence de réseaux de régulation adaptées pour traiter des données — continues ou non — d'expression de gènes.

Ces méthodes ont prouvé une certaine capacité à inférer des réseaux de régulations fiables, cependant elles souffrent toutes de plusieurs contraintes inhérentes au traitement des données d'expression de gènes à savoir : une taille réduite d'échantillons, une dimensionnalité élevée des données et un niveaux de bruit élevé. Afin d'atténuer l'impact négatif de ces contraintes, une nouvelle famille d'algorithmes exploitant les méthodes d'ensemble — parfaitement adaptées à ce scénario — a connu un succès croissant dans le domaine des réseaux de régulation. Ces derniers ont démontré leur efficacité à prédire les interactions et seront décrites en détail dans le prochain chapitre.

1. Les challenges Dream sont détaillés ultérieurement dans l'annexe B

Chapitre 2

Les méthodes d'ensemble

Sommaire

Ce chapitre présente les méthodes d'ensemble : une technique combinant plusieurs modèles de prédiction afin de construire un modèle plus *précis* en terme de prédiction. Ces approches ont des fondements théoriques expliquant ces bonnes performances, citons comme exemple la réduction du *biais/variance* de l'erreur de généralisation résultant du processus d'agrégation. De plus, ces méthodes sont particulièrement adaptées aux contraintes d'inférence des réseaux de régulation, à savoir la grande dimensionnalité des données et le niveau élevé du bruit. Le chapitre est organisé comme suit : la Section 2.1 expose le principe général des méthodes d'ensemble. Ensuite, nous présentons différentes méthodes de construction d'ensemble. La section 2.4 est consacrée à la combinaison des modèles pour obtenir une prédiction d'ensemble. Par la suite, la section 2.5 passe en revue les algorithmes qui peuvent être utilisés pour mettre en œuvre cette technique d'ensemble. La section 2.6 détaille les différents processus de sélection au sein de l'ensemble. Enfin, la section 2.7 décrit l'application des méthodes d'ensemble à l'inférence des réseaux de régulation.

2.1 Principe général

Le but principal d'une méthode d'apprentissage supervisé est d'apprendre un classifieur $f(x)$ pour la variable de classe (discrète ou continue) y lorsque seul le vecteur x est observé. Un classifieur de y est une hypothèse f qui produit une prédiction de cette variable suivant x. Soit f_0 l'hypothèse optimale inconnue qui génère y à partir de x, $f_0(x) = y$. Une estimation de f_0, notée $\hat{f}$, est un modèle automatiquement induit à partir des données d'apprentissage $\mathcal{D}$ [BN06, HTLF01]. Ce modèle suppose que f_0 peut être correctement approximé par une hypothèse appartenant à un espace restreint de candidats $\mathcal{F}$. Le processus d'approximation consiste à estimer certains paramètres θ qui identifient f au sein de $\mathcal{F}$. L'estimation de θ est réalisée de sorte que les prédictions de f sont les plus proches possibles de la prédiction de f_0 pour les instances de $\mathcal{D}$. Cependant, le processus d'approximation doit être mis en œuvre avec soin pour s'assurer que les prédictions de f sont également proches de celles de f_0 pour de nouveaux cas non inclus dans $\mathcal{D}$. Lorsque ceci est réalisé, nous disons que f possède des propriétés de généralisation performantes.

D'une manière plus formelle, les notations spécifiées pour les entités que nous allons utiliser tout au long de ce chapitre sont les suivantes :

Notations

- $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$; $\mathcal{D}$ est l'ensemble des données d'apprentissage contenant n vecteurs x. Une variable prédite y est associée à chaque x.
- $f(x) = y$; f est une hypothèse qui produit une prédiction y à partir du vecteur d'attributs x.
- f_0 est l'hypothèse optimale à approximer
- $\hat{f}$ est l'hypothèse construite à partir de $\mathcal{D}$ qui approxime l'hypothèse f_0
- $\mathcal{F}$ est l'espace des hypothèses possibles
- θ constitue les paramètres d'approximation permettant d'identifier f au sein de $\mathcal{F}$.

L'inférence de modèles de prédiction à partir d'une quantité limitée de données bruitées est un problème difficile. En particulier, il y a une certaine incertitude quant à la forme du modèle qui devrait être utilisée pour représenter les données observées. En outre, il est également difficile d'estimer les valeurs exactes des paramètres du modèle par rapport à ces données. Les méthodes d'ensemble peuvent être utilisées pour résoudre ces difficultés. En effet, ce paradigme d'apprentissage construit une collection de modèles de prédiction dont les réponses individuelles sont ensuite combinées pour prédire la classe de nouvelles instances dans le but d'obtenir un classifieur plus robuste [Die97, HS90, Kun07, MO11, SHA96, WYH03].

Les différents classifieurs de l'ensemble peuvent être produits par une même méthode d'apprentissage à partir de données d'apprentissage de différente distribution de probabilité, *i.e.*, ensembles homogènes [Bre96a, **?**, FS+96]. Ils peuvent égalementêtre produits par différentes méthodes d'apprentissage, à partir d'un unique échantillon d'apprentissage, *i.e.*, ensembles hétérogènes [CNCK04, BN00, TKV04].
La combinaison de plusieurs modèles améliore la précision dans les problèmes de régression et de classification. Plus précisément, les performances de généralisation de l'ensemble sont souvent supérieures à celles du meilleur de l'ensemble [BK99, Bre96a, Die00b, FS+96,

Fri01, MO11, Web00]. D'une part, ces améliorations de performances résultent de la combinaison de classifieurs précis dont les erreurs sont complémentaires [Die00a, FR05, HS90, SS97, UN96]. Autrement dit, la combinaison de classifieurs similaires ne conduit pas à un gain de précision de la prédiction [UN96]. D'autre part, la combinaison de classifieurs faibles, dont la précision est faiblement supérieure à celle d'un classifieur aléatoire, peut conduire à une dégradation significative de la performance de l'ensemble [HS90]. Par conséquent, les méthodes d'ensemble tentent de générer des classifieurs qui sont à la fois précis et divers (qui ne se trompent pas dans les mêmes instances de données). Ceci peut être obtenu par l'apprentissage de chaque prédicteur de l'ensemble, en utilisant une version perturbée de données d'apprentissage [Bre96a, FS$^+$96, Ho98, MMS05] ou par l'introduction d'aléa dans le processus d'apprentissage des membres de l'ensemble [Bre01, GEW06].

Dans le but d'avoir une décision finale d'ensemble, il est nécessaire de combiner les prédictions individuelles des membres de l'ensemble. Dans la pratique, différents algorithmes peuvent être utilisés à cette fin [BN00, Has97, Kit98]. Néanmoins, dans de nombreuses méthodes d'ensemble, le processus de combinaison est très simple [Bre96a, Bre01, GEW06, LY99, MMS05, RIA06]. Dans les problèmes de régression, la moyenne des prédictions des éléments de l'ensemble est généralement utilisée en tant que prédiction de l'ensemble. Par contre, dans les problèmes de classification, c'est plutôt le vote majoritaire qui est utilisé et représente la prédiction la plus fréquente parmi les prédictions des membres de l'ensemble. Le vote majoritaire ou encore la moyenne des prédiction, sont deux méthodes de combinaison qui sont très robustes [FR05, KHDM98, IWAD03, LS97, TG96, UN96]. Il existe d'autres méthodes de combinaison, qui emploient des fonctions de décision non linéaires ou placent les éléments de l'ensemble dans des structures d'arborescence complexes [GB00, IJ94, WBD$^+$01]. Cependant, il n'existe aucune preuve solide qui confirme que l'utilisation de combinaisons complexes aboutit à une meilleure performance en généralisation.

2.2 Pourquoi utiliser les méthodes d'ensembles ?

Dans les applications des méthodes d'apprentissage, un algorithme donné est souvent jugé meilleur que d'autres par rapport à une tâche d'apprentissage bien spécifique. Autrement dit, aucune méthode d'apprentissage ne peut réaliser une performance de généralisation meilleure que toutes les autres méthodes dans toutes les tâches de prédiction possibles [Sch94, Wol96]. La supériorité apparente d'un algorithme d'apprentissage est uniquement dûe à la nature des problèmes étudiés et/ou à la distribution des données.

En dépit de ces considérations, les méthodes ensemblistes ont montré une excellente performance dans de nombreuses tâches d'apprentissage d'intérêt pratique [HS90, KJS94, PC92, XKS92, HTIWG10, DMSES12, SA12, HMVLV12]. En particulier, les erreurs de prédiction d'un membre de l'ensemble peuvent être compensées par les décisions des autres membres. Cette idée intuitive est illustrée par un simple exemple décrit par Dietterich [Die00a], dans le cas d'un problème de classification.

Considérons une tâche de classification binaire. Supposons que les classifieurs d'un ensemble de taille M font des erreurs indépendantes avec une probabilité commune $p < \frac{1}{2}$ et que les prédictions de ces classifieurs sont combinées par un simple vote à la majorité,

c'est à dire, la prédiction qui reçoit le plus grand nombre de votes est la prédiction finale de l'ensemble.

$$Err_{ens}(p, M) = \sum_{m=\left[\frac{M}{2}\right]}^{M} (\binom{M}{m}\, m^m (1-p)^{M-m} = I_p(\left[\frac{M}{2}\right] + 1, M - \left[\frac{m}{2}\right]) \tag{2.1}$$

L'erreur globale de cet ensemble, notée $Err_{ens}(p, M)$, est donnée par la probabilité que plus de la moitié des classifieurs prédisent simultanément d'une manière erronée, où $I_x(a, b)$ est la fonction bêta de régularisation incomplète [AS64].

D'après l'équation 2.1, pour différentes valeurs de M et p, $Err_{ens}(p, M)$ change de comportement.
En effet, $Err_{ens}(p, M) = p$ pour $M = 1$. Cependant, si $p < \frac{1}{2}$ et pour des grandes valeurs de M, l'$Err_{ens}(p, M) << p$. Ainsi, la combinaison des différents classifieurs réduit l'erreur de prédiction. Les principes clés de cette analyse sont l'indépendance des erreurs des membres de l'ensemble et les probabilités d'erreur individuelles p, $p < \frac{1}{2}$.
Dans le cas où $p > \frac{1}{2}$, l'erreur de l'ensemble se comporte différemment. Le processus d'agrégation par vote à la majorité a l'effet inverse. En effet, il augmente l'erreur de prédiction d'un seul membre et pour des grandes valeurs de M l'$Err_{ens}(p, M) >> p$.

Cette simple analyse soulève une question importante dans les méthodes d'ensemble. L'efficacité de l'ensemble repose à la fois sur la précision individuelle des différents membres de l'ensemble et l'indépendance de leurs erreurs qui n'est autre que la diversité entre les membre de l'ensemble [Die00a, HS90, UN96]. Idéalement, les membres de l'ensemble doivent faire mieux que le hasard et leurs erreurs doivent être non corrélées. De telles propriétés dans les membres engendrent une complémentarité dans l'ensemble.

L'exemple précédent confirme que la construction d'ensembles de classifieurs est bénéfique. Toutefois, l'hypothèse des erreurs non corrélées ne peut pas être toujours satisfaite dans la pratique. Les classifieurs se trompent souvent sur les mêmes cas. En dépit de cette limitation, comme décrit par Dietterich [Die00a], il y a plusieurs raisons qui expliquent pourquoi l'agrégation de différents classifieurs peut être un moyen efficace pour améliorer les performances de généralisation.

La première raison est d'ordre statistique. En particulier, presque tous les algorithmes d'apprentissage effectuent une recherche dans l'espace $\mathcal{F}$ des hypothèses possible $\hat{f}$. La tâche de l'algorithme d'apprentissage est de trouver l'hypothèse dans cet espace qui correspond le mieux aux données observées. Si la quantité de données d'apprentissage est limitée, il peut y avoir différentes hypothèses dans $\mathcal{F}$ avec des performances semblables sur ces données. L'agrégation de ces hypothèses dans un ensemble, réduit le risque d'en sélectionner une mauvaise.

La deuxième raison est algorithmique. Comme décrit plus haut, l'algorithme d'apprentissage effectue une recherche dans l'espace $\mathcal{F}$ des hypothèses candidates $\hat{f}$ pour trouver une bonne estimation de f_0. Dans cette recherche, l'algorithme peut être piégé dans un minimum local, qui est une solution sous optimale [HTF09]. L'agrégation de plusieurs hypothèses $\hat{f}$, obtenues suite à la répétition de processus de recherche partant de différents points, peut aboutir à une meilleure approximation de la cible f_0 que toutes

autres hypothèses simples f.

La troisième raison est liée à la capacité de représentation élargie des ensembles. Dans de nombreuses applications d'apprentissage, la cible f_0 ne peut pas être bien estimée par les hypothèse de $\mathcal{F}$. A titre d'exemple, un problème de classification demandant une décision non-linéaire ne peut être résolu par un modèle linéaire simple. Toutefois, en combinant les sorties des différents modèles linéaires, il est possible d'obtenir une décision non linéaire. Les méthodes d'ensemble peuvent donc être utilisées pour surmonter le problème décrit. En particulier, en agrégeant les hypothèses $\hat{f}$ dans $\mathcal{F}$, il est possible d'élargir l'espace d'hypothèses candidates, afin d'obtenir un modèle plus expressif.

En plus de ces raisons, nombreuses preuves montrent que la combinaison des prédictions des membres de l'ensemble réduit la variance de l'erreur en généralisation [Bre96a, GEW06, UN96] et dans certains cas également du biais [Bre98, SFBL98, WB98]. En outre, certaines méthodes d'ensemble telles que le bagging ou les forêts aléatoires sont très robustes à la présence d'instances bruitées dans les données d'apprentissage [Bre01, Die00b, MMHLS09, MO11].

2.3 Méthodes de construction des classifieurs

Différentes techniques ont été proposées pour construire les classifieurs de l'ensemble [BWHY05, Die00a, Kun07, BDP06, Kun07, SHA96, VM02]. Celles-ci peuvent se regrouper dans les catégories suivantes.

2.3.1 La randomisation dans les données d'apprentissage

Il s'agit de la stratégie la plus fréquemment utilisée pour construire les classifieurs de l'ensemble. Elle consiste à perturber les données d'apprentissage initiales en supprimant ou en ajoutant une partie de données ou encore en modifiant leurs poids relatifs dans l'ensemble d'apprentissage. La diversité entre les classifieurs dans l'ensemble est obtenue grâce à l'apprentissage de chacun d'eux sur une version perturbée différente des données initiales. Pour que cette méthode soit efficace, les classifieurs individuels doivent présenter quelques instabilités. Cela signifie que de petits changements dans les données d'apprentissage doivent conduire à des changements dans les prédictions des modèles appris. Les réseaux de neurones et les arbres de décision sont des modèles d'apprentissage connus pour avoir cette propriété [Bre98]. Par conséquent, ils sont de bons candidats pour construire des ensembles à l'aide de ce mécanisme.

En revanche, les modèles linéaires, les SVM [Vap00] et les K-plus proches voisins [HTLF01] sont relativement stables [Bre98]. Utiliser cette stratégie avec les modèles stables ne devrait pas être utile. Néanmoins, il existe certaines études [KPJ+03, SD02] montrant que si les modifications de l'ensemble d'apprentissage sont suffisamment grandes, ces modèles deviennent instables.

Une méthode qui peut être utilisée pour perturber des données d'apprentissage est le *bootstrap* [ET94]. Les échantillons bootstrap sont obtenus par un tirage avec remise des données. Un bootstrap de même taille que l'ensemble d'apprentissage contient en moyenne seulement 63,2 % de cas différents, les autres données étant des instances répétées

[ET94]. Ainsi, les échantillons de bootstrap formés sont très différents les uns des autres car, en moyenne, ils partagent seulement 39,9% de leurs instances. Le bagging [Bre96a] et forêts aléatoires [Bre01] sont deux exemples de méthodes d'ensemble, qui utilisent des échantillons bootstrap pour construire les différents classifieurs de l'ensemble.

Au lieu de supprimer des instances de données de l'ensemble d'apprentissage, d'autres méthodes attribuent des poids différents aux différentes instances. Cette procédure est utilisée pour le boosting pour avoir des classifieurs complémentaires [Dru97, FS95, Fri01, MR03]. Dans le boosting, les membres de l'ensemble sont générés de façon séquentielle. Le premier classifieur est obtenu à l'aide de l'affectation de poids égaux aux instances, Ensuite, les classifieurs ultérieurs sont générés en modifiant les instances d'apprentissage. Les poids les plus importants sont affectés aux instances qui ne sont pas correctement prédites par les classifieurs générés dans la séquence. En revanche, le poids des instances correctement prédites sont diminuées. Ce mécanisme a montré qu'il génère un ensemble de classifieurs très divers [Die00b]. D'autres méthodes d'ensemble générant de la diversité en manipulant les instances de données d'apprentissage existent, citons le subagging et le bragging [**?**], le wagging [BK99], le multiboosting [Web00] ou encore le bag-boosting [Det04].

2.3.2 La randomisation dans l'algorithme d'apprentissage

Une autre technique pour la construction de méthodes d'ensemble consiste à randomiser l'algorithme d'apprentissage des membres de l'ensemble. En conséquence, différents classifieurs vont être produits, ce qui accroît la diversité de l'ensemble. Cette technique doit être utilisée avec précaution, car l'introduction de l'aléatoire dans l'algorithme d'apprentissage peut parfois conduire à une baisse significative de la précision des classifieurs obtenus.

Introduire l'aléatoire dans l'algorithme d'apprentissage peut être réalisé de différentes manières. Par exemple, l'algorithme d'apprentissage C4.5 pour la construction d'arbres de décision peut être rendue aléatoire. Au lieu de choisir à chaque nœud la meilleure partition possible en fonction de certains critères, Dietterich et Kong [DK95] proposent de choisir au hasard parmi les s meilleures partitions, avec $s = 20$.

Une procédure de randomisation similaire est mise en œuvre dans les forêts aléatoires. Dans ce cas, le meilleur attribut de partition est sélectionné à chaque noeud parmi un sous-ensemble aléatoire de m variables, où m est généralement fixé à $[\log_2(k) + 1]$ ou $\left\lfloor \sqrt{k} \right\rfloor$, avec k le nombre de variables en l'entrée [BHA09a, Bre01]. D'autres randomisations sont mises en oeuvre dans le processus de construction des arbres [GEW06]. Par exemple, la partition des nœuds internes est réalisée en utilisant des attributs aléatoires et des seuils de partition aléatoires.

Dans ces techniques, la quantité de randomisation introduite dans l'algorithme d'apprentissage est spécifiée par un paramètre (*e.g.* m dans le cas des forêts aléatoires). La performance de l'ensemble dépend en fait de ce paramètre [BHA09a, GEW06] qui doit être choisi avec précaution afin d'atteindre la meilleure précision de prédiction. Dans plusieurs cas, utiliser des règles simples pour fixer ce paramètre est suffisant [Bre01, BHA09a].

2.3.3 La manipulation des variables d'entrée

Cette technique élimine sélectivement une partie des variables d'entrée contenue dans le vecteur de l'attribut x des données d'apprentissages. La mise en œuvre de ce processus n'est pas simple, car la suppression de certains attributs peut conduire à une diminution spectaculaire de la précision des classifieurs résultants [TG96]. La méthode des sous-espaces aléatoires est un exemple d'algorithme d'ensemble qui utilise cette technique pour générer des classifieurs individuels [Ho98]. Dans cette méthode, chaque classifieur est généré en utilisant uniquement un sous-ensemble de variable choisi aléatoirement à partir du vecteur initial d'attributs x. La méthode des sous-espaces aléatoires génère des classifieurs qui peuvent être aussi divers que ceux produits par le boosting [Ho98].

L'idée d'utiliser uniquement un sous-ensemble d'attributs a également été employée dans une méthode appelée bagging d'attributs [Büh12]. Au lieu de sélectionner un sous-ensemble de variables d'entrée, de nouvelles variables d'entrée sont générées. Par exemple, les nouvelles variables peuvent être obtenues en projetant l'espace des variables originales dans un nouvel espace, et ce en utilisant la technique de projection d'analyse en composantes principales (ACP). En effet, Skurichina et Duin [SD05] constatent que les ensembles construits avec cette méthode donnent de meilleurs résultats que les ensembles générés avec la sélection aléatoire de variables. La technique de projection ACP est également utilisée dans l'algorithme d'apprentissage des forêts de rotation [RIA06]. Par ailleurs, il existe des opérateurs de projection autre que l'ACP qui peuvent être utilisés pour générer de nouvelles variables d'entrée [Che96, DT00, FB03].

2.3.4 La manipulation des variables cibles

Dans cette technique, la variable cible associée à chaque vecteur d'attributs x dans les données d'apprentissage est perturbée avant l'apprentissage des classifieurs de l'ensemble.

Le codage correcteur d'erreurs en sortie (ECOC) est un exemple de méthode d'ensemble de classification qui utilise cette technique pour générer les classifieurs de l'ensemble [DB95]. Ainsi, avant l'apprentissage du j^{eme} membre de l'ensemble, cette méthode partitionne aléatoirement l'ensemble des différentes étiquettes des classes en deux ensembles disjoints A_j et B_j. De nouvelles étiquettes sont ensuite affectées aux instances d'apprentissage, et ce en fonction de leur appartenance à l'un de ces deux ensembles. Le j^{eme} classifieur est alors construit sur les données avec ces nouvelles étiquettes. Une fois que l'ensemble est généré, la prédiction des étiquettes de classes sur des instances de test est calculée en utilisant le vote à la majorité, où le j^{eme} classifieur vote pour toutes les étiquettes de classe incluses dans A_j ou B_j. L'évaluation expérimentale de cette méthode montre qu'elle améliore les performances d'un unique arbre de classification et d'un unique réseau de neurones sur plusieurs problèmes multi-classes [DB95]. En outre, Schapire [Sch97] a montré que la combinaison d'ECOC avec le boosting est également efficace pour les problèmes multi-classes. Une limitation de la méthode ECOC est que le nombre d'étiquettes différentes de classes doit être grande pour obtenir des améliorations significatives.

Au lieu de ré-étiqueter les instances d'apprentissage, d'autres techniques injectent aléatoirement du bruit dans les étiquettes de classe. Cette idée est employée dans [Bre00] pour construire des ensembles fondés sur la randomisation au niveau de la sortie. Dans les problèmes de régression, un bruit gaussien est injecté dans la variable cible de chaque

instance avant l'apprentissage des classifieurs de l'ensemble. Dans les problèmes de classification, les étiquettes de classe sont modifiées de manière aléatoire. La modification est réalisée de telle sorte que les proportions des classes dans les données d'apprentissage sont préservées. Cette technique est gérée par un paramètre d'entrée qui doit être réglé à sa valeur optimale. Les résultats expérimentaux montrent que les deux méthodes ont de meilleures performances que le bagging [Bre00].

Un autre algorithme d'apprentissage d'ensemble qui injecte du bruit dans les étiquettes de classe est la commutation de classe (class-switching) [GMMS08, MMS05]. Contrairement à la méthode précédente, la commutation de classe ne préserve pas la distribution d'origine de classe dans les données d'apprentissage perturbées. Ainsi, dans le cas de déséquilibre des étiquettes de classe, le paramètre qui gère le taux du changement peut prendre des valeurs plus grandes que celles de la méthode précédente. Généralement, cette technique donne de meilleurs résultats que ceux de la méthode précedente [MMS05].

2.4 Méthodes pour combiner les membres de l'ensemble

Dans cette partie, nous examinons quelques méthodes qui ont été proposées pour combiner les prédictions des membres de l'ensemble afin d'avoir une prédiction finale. Ces méthodes peuvent être regroupées en plusieurs catégories [KDM00]. Nous nous intéressons aux combinaisons les plus simples, les combinaisons en parallèle.

Dans la combinaison en parallèle, les classifieurs de l'ensemble sont interrogés de manière indépendante et leurs réponses sont ensuite combinées. Par exemple, pour le vote à la majorité, les différents classifieurs prédisent un état, la décision finale de l'ensemble est la prédiction qui reçoit le plus de votes. Cette méthode est employée par plusieurs méthodes comme le bagging ou les forêts aléatoires. Dans les problèmes de régression, la prédiction finale de l'ensemble est souvent calculée par la moyenne des prédictions des différents membres de l'ensemble. Ces deux méthodes de combinaison sont très robustes [FR05, KHDM98, IWAD03, LS97, TG96, UN96]. Entre autres, au lieu de considérer le même poids pour chaque classifieur, ces deux méthodes peuvent également attribuer des poids différents aux différents classifieurs dans l'ensemble. En effet, le vote à la majorité pondérée est employé dans les algorithmes de boosting comme Adaboost [FS$^+$96]. La pondération de la moyenne est aussi souvent utilisée dans les ensembles de régression [Has97, PC92].

Un autre exemple de méthode de combinaison se trouve dans le paradigme de mélanges d'experts [AINH91]. Après avoir interrogé les différents classifieurs (experts) de l'ensemble (mélange), cette méthode calcule une combinaison linéaire de leurs prédictions.

Au lieu d'utiliser une combinaison linéaire, il existe plusieurs méthodes qui emploient des fonctions non linéaires pour générer la prédiction finale d'ensemble, comme le stacking [WB98] qui est un procédé de combinaison non-linéaire utilisant en entrée les sorties des différents membres de l'ensemble d'apprentissage. Cette méthode peut déduire les biais des classifieurs de l'ensemble. Le stacking a été appliqué à des tâches de régression dans [Bre96b] et a été étendu à d'autres algorithmes [TD03].

Enfin, il existe d'autres méthodes qui peuvent être utilisées pour combiner les sorties

des différents membres de l'ensemble, à savoir la combinaison naïve Bayes, combinaison multi nominale, l'intégrale floue, etc. Voir [Kun07] pour une description de ces méthodes et [ICD01] pour une comparaison empirique.

2.5 Les méthodes d'ensemble les plus populaires

Dans cette partie, nous décrivons les algorithmes les plus représentatifs des méthodes d'ensemble, *i.e.*, le bagging, le boosting et les forêts aléatoires. Nous détaillons la méthode de bagging [Bre96a] plus que les autres, car c'est la méthode que nous utilisons par la suite.

2.5.1 Le bagging

Le bagging est une méthode d'ensemble simple dans lequel les classifieurs individuels sont de même type et sont construits sur différents échantillons bootstrap à partir des données d'apprentissage [Bre96a]. Ainsi, nous commençons par décrire les échantillons bootstrap [ET94]. Ensuite, nous montrons comment cette approche obtient de meilleures prédictions en agrégeant les sorties des membres de l'ensemble. L'origine de cette amélioration est dû à la réduction de la variance de l'erreur en généralisation.

Les échantillons boostraps L'échantillonnage d'un bootstrap est une méthode qui fournit un moyen de calcul direct pour évaluer l'incertitude d'estimation des paramètres d'un modèle [ET94, HTLF01]. Cette technique suppose que l'incertitude provient du fait de considérer que les données observées $\mathcal{D}$ sont une réalisation d'une variable aléatoire. Supposons que θ est le paramètre d'un modèle et que les données observées $\mathcal{D}$ ont été générées par ce modèle pour une valeur de θ, notée θ_0, Une estimation de θ_0, notée $\hat{\theta}$, peut être obtenue en minimisant une fonction de perte L qui dépend de $\mathcal{D}$ et θ.

$$\theta = arg_\theta \; min \; L(\theta|\mathcal{D}) \tag{2.2}$$

Les choix courants pour L sont la somme des erreurs quadratiques ou de l'entropie croisée [ET94]. Sachant que les données $\mathcal{D}$ utilisées pour effectuer l'estimation sont considérées comme des réalisations indépendantes de variables aléatoires, $\hat{\theta}_0$ est par conséquent une variable aléatoire qui dépend de $\mathcal{D}$.

La méthode de bootstrap fournit un moyen d'approcher la distribution sous-jacente de l'estimation de $\hat{\theta}$ sans connaissance de la vraie distribution de $\mathcal{D}$. Pour cette raison, cette technique construit un ensemble de B jeux de données $\mathcal{D}_b, b = 1, ..., B$. Chaque $\mathcal{D}_b$ de données est un échantillon bootstrap construit par des tirages de $\mathcal{D}$ avec remise. Ces échantillons bootstrap peuvent être considérés comme des versions modifiées des données originales $\mathcal{D}$. La distribution de probabilité de $\hat{\theta}_0$ peut être approximée par le calcul d'un ensemble de réplicat de boostrap $\{\hat{\theta}_0^{(b)}, i = 1, ..., B\}$, où chaque réplicat $\hat{\theta}_0^{(b)}$ est obtenu suivant l'équation 2.2, et ce en utilisant $\mathcal{D}_b$ au lieu de $\mathcal{D}$.

Sous certaines conditions, la méthode bootstrap est asymptotiquement stable [ET94]. En particulier, tant que la taille des données observées augmente, la distribution empirique des données observées dans $\mathcal{D}$ approche asymptotiquement de la vraie distribution des données. Cependant, la méthode ne garantit pas des échantillons finis et elle peut échouer

quand il y a une inadéquation entre la vraie distribution des données observées et la distribution empirique [ET94]. Le principal avantage de la méthode bootstrap par rapport aux méthodes analytiques est qu'elle est très simple à mettre en oeuvre. Elle peut être appliquée à des estimateurs très complexes pour lesquels il peut être difficile de tirer des formules exactes de moyennes, d'écarts-types ou d'intervalles de confiance [ET94].

Le nombre B de réplicats de bootstrap est souvent déterminé par les ressources de calcul disponibles. Typiquement, quelques centaines de répétitions sont suffisantes. Il existe, par ailleurs, une procédure similaire au boostrap, le Subsampling [NPW99], qui est basée sur l'échantillonnage sans remise au lieu de celui avec remise.

L'agrégation des boostraps Le bagging est un acronyme pour *bootstrap aggregation* [Bre96a]. Comme décrit dans la partie précédente, la méthode bootstrap peut être utilisée pour déterminer l'incertitude dans l'estimation des paramètres du modèle à partir d'échantillons de taille finie. Dans cette partie, nous montrons que le bootstrap peut être également utilisé pour améliorer la prédiction. En terme de prédiction, ces améliorations peuvent être expliquées par la réduction de la variance de l'erreur de prédiction [Bre98].

Dans la pratique, nous disposons d'un seul jeu de données $\mathcal{D}$ pour l'apprentissage. Le bagging pallie à cette limitation, en générant des ensembles de données à partir de différents échantillons boostrap qui servent dans la construction des classifieurs. L'inconvénient de cette méthode est qu'elle introduit des corrélations entre les classifieurs, ce qui met en cause le critère d'indépendance. De plus, l'utilisation des échantillons de bootstrap au lieu de $\mathcal{D}$ rend le biais et la variance des membres de l'ensemble appris légèrement plus élevés que ceux d'un classifieur formé avec toutes les données disponibles dans $\mathcal{D}$. Néanmoins, l'estimation agrégée résultante a souvent une meilleure performance de généralisation qu'un seul [BK99, Bre96a, MO11]. L'algorithme 1 illustre le pseudo algorithme du bagging.

Algorithme 1 pseudo algorithme du bagging

Entrées : $\mathcal{D}$: les données d'apprentissage, M : les données d'apprentissage
Sorties : $\hat{f}_{bag}(x)$: les estimations agrégées
1 : **Pour tout** $i \in 1..M$ **Faire**
2 : Construction des échantillons boostrap $\mathcal{D}_i$ de $\mathcal{D}$
3 : Apprentissage du modèle $\hat{f}_i(x)$ à partir de $\mathcal{D}_i$
4 : **Fin pour**
5 : Agréger les estimations des modèles : $\hat{f}_{bag}(x) = \frac{1}{M}\sum_{i=1}^{M}\hat{f}_i(x)$

2.5.2 Les forêts aléatoires

Initialement inspirées par le travail de Amit et Geman [AGW97], les forêts aléatoires ont été introduites par Breiman [Bre01] comme une amélioration du bagging lorsque les membres de l'ensemble sont des arbres. Les forêts aléatoires peuvent être considérées comme un algorithme de bagging où les arbres sont remplacés par des arbres construits avec une version aléatoire de l'algorithme CART [BFOS84]. L'idée principale est de générer un ensemble de M $\{h_1, .., h_M\}$ arbres doublement perturbés au moyen d'une randomisation opérée à la fois au niveau de l'échantillon d'apprentissage et à la méthode de construction

des nœuds internes des arbres. Une forêt aléatoire peut former un ensemble d'arbres de décision ou de régression selon la problématique.

Chaque arbre de l'ensemble est ainsi généré au départ sur un sous-échantillon bootstrap $\mathcal{D}_{i(i=1,\ldots,M)}$ des données d'apprentissage $\mathcal{D}$, de manière similaire aux techniques de bagging. Ensuite, l'arbre est construit, mais à chaque nœud, la sélection du meilleur attribut pour la partition s'effectue non pas sur l'ensemble complet des N attributs mais sur un sous-ensemble sélectionné aléatoirement au sein de celui-ci. Le but de l'introduction de l'aléatoire dans la sélection des attributs dans la partition est d'accroître la diversité entres les arbres générés. La taille F de cette sélection est généralement fixée à $\sqrt{N}$ afin d'assurer s'équilibre entre le biais et la variance. Chaque arbre de la forêt est ainsi développé jusqu'à sa taille maximale, sans élagage. Les prédictions des arbres sont ensuite combinées par un simple vote à la majorité, dans le cas de classification, et par le calcul de la moyenne, dans le cas de la régression. Le principal avantage de cette structure est qu'elle permet d'éviter le danger que représente le sur-apprentissage pour toute méthode de prédiction.

Breiman [Bre01] prouve que la clé de performance de prédiction des forêts consiste dans la production d'un ensemble d'arbres peu corrélés, tout en préservant autant que possible leur qualité individuelle. Cet objectif est atteint grâce à la double randomisation, d'une part, par la technique du bagging ayant déjà prouvé son efficacité [Bre96a, BK99, Die00b], d'autre part, par la perturbation supplémentaire opérée au niveau du choix des partitions optimales à l'intérieur même de l'arbre. Les procédures de sélection sont maintenues sur le sous-ensemble d'attributs échantillonnés à chaque noeud.

Out-Of-Bag Parallèlement à ses excellents résultats en prédiction, la structure des forêts aléatoires permet de livrer des renseignements complémentaires concernant l'estimateur qu'elle construit. L'utilisation d'échantillons bootstrap permet de n'utiliser en moyenne que 66% des individus de la base d'apprentissage et autorise notamment le calcul du taux d'erreur Out-Of-Bag (OOB) représenté par l'ensemble des erreurs de classification sur le reste (33%) des individus non consommés dans la construction de chaque modèle d'apprentissage. En effet, ces individus sont propagés uniquement dans les arbres où ils ne sont pas utilisés dans la phase de la construction. Ceci fournit l'erreur OOB qui est une estimation non biaisée du taux d'erreur en généralisation sans avoir recours à un échantillon de test supplémentaire.

Paramétrages Ces considérations théoriques ont été vérifiées empiriquement par Breiman [Bre01] sur une série de 20 jeux de données (16 réels et 4 artificiels). Il apparait en outre dans cette expérience que le nombre F d'attributs pré-sélectionnés aléatoirement pour la construction de chaque nœud a peu d'influence sur le taux d'erreur en généralisation final. Breiman recommande d'utiliser $\sqrt{M}$ (M est le nombre total d'attribut) comme la valeur par défaut de F qui assure l'équilibre entre le biais et la variance [Bre01]. D'un autre côté, il a prouvé que lorsque le nombre d'arbres E impliqués dans la forêt de prédiction augmente, le taux d'erreur en généralisation converge vers une valeur limite.

Non sensibilité au bruit Dans [Bre01], Breiman a également testé l'effet de l'addition de bruit sur les prédictions en généralisation sur 9 jeux de données réels. Contrairement aux techniques de boosting, les performances des forêts aléatoires ne sont que peu dégradées

par l'adjonction de 5% de bruit aléatoire, ce qui confirme les résultats déjà obtenus avec les techniques de bagging simple.

En outre, deux autres informations peuvent être calculées par les forêts aléatoires, une mesure de l'importance des différentes variables dans le classifieur final, et une mesure de proximité entres les individus.

Mesure d'importance des variables L'importance des variables est une notion difficile à définir, celle-ci pouvant être liée à des interactions complexes dans la structure du concept. Ce paramètre peut être estimé par deux mesures distinctes :

- La première mesure consiste à calculer l'augmentation du taux d'erreur OOB lorsque les modalités de chaque variable étudiée sont permutées aléatoirement sur les individus OOB, les autres variables restant inchangées.
- La deuxième mesure ne requiert pas d'altération du jeu de données et est évaluée par la décroissance moyenne du critère de Gini dans la forêt directement liée à l'utilisation de la variable en question.

Cette évaluation est plus rapide à obtenir mais moins fiable que la précédente. Ces estimations permettent une meilleure compréhension du concept recherché et peuvent conduire a des présélections de variables, et donc a une réduction dimensionnelle et une simplification des problèmes traités.

Calcul de proximité Les forêts aléatoires fournissent également une matrice de proximité des individus. La proximité entre deux individus est calculée par la fraction d'arbres générés dans lesquels ces derniers appartenant à une même feuille, postulant que deux individus proches devraient suivre un cheminement identique dans l'arbre. Cette mesure peut être utile lors d'une recherche de structure au sein du jeu de données, et ouvre la voie à l'utilisation des forêts aléatoires en classification non supervisée.

En raison de la bonne performance de prédiction des forêts aléatoires sur des données de grande dimension, de nombreuses variantes ont été développées. Par exemple, Geurts et al. [GEW06] ont proposé une méthode d'ensemble d'arbre appelé *extra-arbres* qui sélectionne à chaque nœud le meilleur attribut parmi F sélectionnés aléatoirement. Contrairement à la version originale des forêts aléatoires apprises avec des multiples sous-ensembles de données d'apprentissage (boostraps), les arbres de base de l'extra-arbres sont construits à partir de l'ensemble complet d'apprentissage et la randomisation est opérée explicitement au niveau du découpage des nœuds. Ce dernier est déterminé parmi F découpages aléatoires, chacun étant déterminé en choisissant au hasard une entrée (sans remise) et un seuil.

2.5.3 Le boosting

Le boosting est une méthode générale pour améliorer les performances de n'importe quel algorithme d'apprentissage [FS+96]. Par définition, un algorithme de boosting peut théoriquement transformer un algorithme d'apprentissage basique (appelé « weak learner ») avec des performances un peu meilleures que le hasard en un algorithme performant. Le principe de ces algorithmes de boosting est de re-pondérer à plusieurs reprises les exemples d'apprentissage et de refaire tourner l'algorithme d'apprentissage sur ces exemples re-pondérés. Le poids des exemples mal classés est augmenté tandis que le poids

des biens classés est diminué. Ainsi, le boosting force cet algorithme à concentrer ses efforts d'apprentissage sur les exemples les plus difficiles. L'hypothèse finale est un vote pondéré des différentes hypothèses obtenues à chaque instance de l'algorithme d'apprentissage. Parmi les algorithmes de boosting les plus utilisés nous trouvons 'Adaboost' [FS95].

2.6 Sélection de modèles dans les méthodes ensemblistes

La clé de l'amélioration de la performance des méthodes d'ensemble est la complémentarité des prédictions données par les membres de l'ensemble. Les mesures de précision ou de diversité ne peuvent être utilisées isolément pour améliorer le rendement de l'ensemble. Dietterich [Die00a] affirme qu' "une condition nécessaire et suffisante pour qu'un ensemble de classifieurs soit performant est que quelque soit le classifieur membre choisi, il doit être précis et différent des autres membres". Cette constatation est valide pour les méthodes ensemblistes homogènes [BHA09b] et hétérogènes [CNCK04]. Ainsi, toute amélioration des performances repose sur les concepts de la précision et de la diversité. Par conséquent, la plupart des méthodes de sélection dans les ensembles tentent d'extraire un petit sous-ensemble de classifieurs complémentaires à partir de l'ensemble original, en y éliminant des modèles.

Lors de la sélection d'un sous ensemble de m modèles à partir d'un ensemble de modèles M, l'espace de recherche est de $2^m - 1$ sous-ensembles possibles. La recherche du sous-ensemble optimal est un problème NP-complet [TX00]. Selon Martínez-Muñoz et Suárez, la sélection devient intraitable pour les valeurs de $M > 30$ [MMS06].

Les heuristiques de sélection de classifieurs sont nombreuses, celles qui sélectionnent les classifieurs les plus précis tel que 'choose best' [PY96], ou encore celles qui sélectionnent les couples de classifieurs les plus divers tel que 'Kappa Prunning' [MD97]. D'autre part, il y a celles qui fonctionnent d'une manière séquentielle [CVZ06] tels que 'Forward' et 'Backward' et d'autres qui utilisent l'aléatoire [ZWT02]. Dans cette section, nous dressons une étude de ces méthodes de sélection en les classant suivant leurs algorithmes de recherche.

2.6.1 Les algorithmes randomisés de sélection

Zhou et al. adoptent une méthode qui utilise une partie seulement des modèles à partir d'un ensemble plutôt que tous les modèles [ZWT02]. Leur travail sur des ensembles de réseaux de neurones, appelés *GASEN* (Genetic Algorithm based Selective ENsemble), commence par l'affectation d'un poids aléatoire à chacun des modèles de base. Ensuite, il emploie un algorithme génétique pour faire évoluer ces poids afin de caractériser la contribution des modèles à l'ensemble. Une fois le processus évolutif terminé, les réseaux de neurones dont les poids optimisé sont en dessous d'un certain seuil sont retirés de l'ensemble. La sortie finale est la moyenne des prédictions des réseaux retenus dans l'ensemble. *GASEN* est appliqué sur plusieurs problèmes de régression et de classification. Les résultats empirique sur les problèmes de régression montrent que *GASEN* surpasse le bagging et le boosting en terme de biais et de variance. Par contre, les résultats des problèmes de classification ne sont pas si prometteurs. Suite à ces travaux, l'approche a

été appliquée avec succès par Zhou et Tang pour construire des ensembles d'arbres de décision [ZT03] et par Hernánez et al. dans le boosting [HLHLRTV06].

2.6.2 Les algorithmes séquentiels de sélection

Les algorithmes séquentiels modifient un ensemble, de manière itérative, en ajoutant ou supprimant des modèles. Trois types d'algorithmes de recherche sont utilisés :

- *Forward* : la recherche commence avec un ensemble vide et ajoute successivement un modèle à l'ensemble à chaque itération ;
- *Backward* : la recherche commence avec tous les modèles de l'ensemble et élimine un modèles à chaque itération ;
- *Forward-Backward* : la sélection est basée à la fois sur le forward et le backward.

La sélection forward La sélection forward commence avec un ensemble vide et ajoute, de manière itérative, des modèles dans le but de diminuer l'erreur de prédiction.
Coelho et Von Zuben [CVZ06] décrivent deux algorithmes de sélection forward appelés sélection séquentielle forward avec classement (*FSSwR*) et sélection séquentielle forward (*FSS*). *FSSwR* classe tous les candidats suivant leurs performances sur un ensemble de données de validation. Ensuite, il sélectionne, à chaque itération, le candidat de meilleure performance jusqu'à ce que la performance de l'ensemble sélectionné diminue.

Concernant *FSS*, à chaque fois qu'un nouveau candidat est ajouté à l'ensemble, tous les candidats sont testés et celui qui mène à l'amélioration de la performance de l'ensemble est sélectionné. En l'absence d'un modèle qui améliore les performances de l'ensemble, la sélection s'arrête. Cette approche est également utilisée dans [RGV01].

Partridge et Yates proposent un autre algorithme de sélection forward similaire à *FSS* [PY96]. La principale différence est que le critère de l'inclusion d'un nouveau modèle est la mesure de diversité. Le modèle avec une plus grande diversité que ceux déjà sélectionnés est également inclus dans l'ensemble. La taille de l'ensemble est fixée par un paramètre d'entrée de l'algorithme.

Une autre approche similaire est présentée dans [HLHLRTV06]. A chaque itération, l'algorithme teste tous les modèles non encore sélectionnés, et choisit celui qui réduit le plus l'erreur en généralisation de l'ensemble sur la base d'apprentissage. Des expériences visant à réduire les ensembles générés à l'aide du bagging sont prometteurs.

La sélection backward La sélection backward commence avec tous les modèles de l'ensemble et supprime, de manière itérative, les modèles dans le but de diminuer l'erreur de prédiction. Coelho et Von Zuben décrivent deux algorithmes de sélection backward, la sélection séquentielle backward avec classement (*BSSwR*) et sélection séquentielle backward (*BSS*) [CVZ06]. Dans le premier cas, les candidats sont classés en fonction de leur performance, sur la base d'un ensemble de données de validation (comme dans *FSSwR*). Le moins bon est ensuite enlevé. Si les performances de l'ensemble s'améliorent, le processus de sélection se poursuit. Sinon, le processus s'arrête. *BSS* est liée à *FSS* de la même manière que le *BSSwR* est liée à la *FSSwR*, c'est à dire, il fonctionne comme *FSS* mais en utilisant la sélection backward au lieu de la sélection forward.

La sélection backward a été de même utilisée dans les travaux de Banfield et al. [BHBK05] pour maximiser la précision de l'ensemble en éliminant les classifieurs dont la

contribution à la performance de généralisation, estimée en termes de mesures de précision ou de diversité sur l'ensemble de l'apprentissage, est petite voire négligeable.

La sélection forward-backward mixte Dans les algorithmes forward et backward décrits par Coelho et Von Zuben, à savoir la sélection *FSSwR*, la *FSS*, la *BSSwR* et la *BSS*, le critère d'arrêt suppose que la fonction d'évaluation est monotone [CVZ06]. Toutefois, en pratique, cette supposition ne peut pas être garantie. L'utilisation mixte du forward et backward, vise à éviter les situations où l'amélioraion rapide dans les itérations initiales ne permet pas d'explorer des solutions moins rapides mais de meilleurs résultats finaux. En effet, Moreira et al. [MdJFSJS06] décrivent un algorithme qui commence par sélectionner de façon aléatoire un nombre prédéfini de m modèles. Ensuite, à chaque itération, un pas forward et un pas backward sont effectués. Le pas forward est équivalent à celui utilisé par *FSS*, *i.e.*, il sélectionne le modèle de l'ensemble qui améliore la précision de l'ensemble sélectionné. À cette étape, l'ensemble se compose de $m + 1$ modèles. Dans la deuxième étape, seules m modèles avec les plus grandes précisions de l'ensemble sont gardés, c'est à dire, en pratique, l'un des $m + 1$ modèles est retiré de l'ensemble. Finalement, le processus s'arrête quand le même modèle est choisi dans les deux étapes.

Margineantu et Dietterich présentent un algorithme de sélection appelé sélection avec réduction d'erreur par ajustement arrière [MD97] (en anglais *reduce-error pruning with back fitting*). Cet algorithme est similaire à *FSS* dans ces deux premières itérations. Après la deuxième itération, lors de l'ajout du troisième candidat et ceux qui suivent, une étape d'ajustement arrière est faite. Considérons C_1, C_2 et C_3 les candidats déjà sélectionnés. Tout d'abord il supprime C_1 de l'ensemble et teste l'ajout de chacun des candidats restants $C_{i(i>3)}$ dans l'ensemble. Puis, il répète cette étape pour C_2 et C_3 et en choisit le meilleur. D'autres itérations sont exécutées jusqu'à ce qu'un nombre d'itérations prédéfini par l'utilisateur soit atteint.

Tout ces algorithmes séquentiels de sélection souffrent du risque de sur-apprentissage. C'est pour cette raison que certains travaux sur la sélection séquentielle de type forward [MD97] ont essayé d'éviter ce problème par l'ajout de trois nouveaux procédés comme la sélection avec remise, l'initialisation par un ensemble performant, et le bagging pour la sélection d'ensemble. Entre autres, nous constatons que toutes ces algorithmes ne tiennent pas compte du compromis entre la précision et la diversité.

2.6.3 La sélection par algorithme génétique, optimisation ou test statistique

Les algorithmes génétiques (AGs) ont également été proposés pour la sélection d'un sous-ensemble quasi-optimale à partir d'un ensemble complet [ZT03, ZWT02]. Dans [ZWT02], la sortie de l'ensemble est une moyenne pondérée des sorties de chaque membre de l'ensemble. L'ensemble optimal de poids des membres de l'ensemble se trouve en minimisant une fonction qui permet d'estimer l'erreur de généralisation de l'ensemble. Le problème de minimisation est résolu par un AG standard avec un système de codage en virgule flottante pour les poids des valeurs réelles. Une fois le processus d'estimation des poids terminé, les réseaux de neurones dont les poids sont en dessous d'un certain seuil sont retirés de l'ensemble. La sortie de l'ensemble sélectionné est la moyenne des

prédictions des réseaux retenus dans l'ensemble. Les expériences menées dans [ZWT02] utilisent *AGs* comme une stratégie pour réduire l'ensemble généré par le bagging. Cette approche a été aussi appliquée dans des ensembles produits par le boosting [HLHLRTV06].

Zhang et al. [ZBS06] propose une nouvelle méthode de la sélection de sous-ensembles fondé sur la programmation semi-définie (en anglais *SemiDefinite Programming SDP*). Dans les tâches de classification, le problème est formulé en termes de $M * M$ matrice G, dont les termes diagonaux G_{ii} mesurent les erreurs individuelles des membres de l'ensemble et dont les termes non diagonaux G_{ij} , $i \neq j$ mesurent le nombre d'erreurs communes entre les classifieurs i et j. L'objectif est alors de trouver la sous-matrice de G, de dimensions $u * u$, correspondant à un sous-ensemble de taille u qui minimise la somme des éléments de la sous-matrice de G. C'est un problème classique d'optimisation qui est aussi NP-difficile.
Le problème peut être reformulé comme un problème de coupe maximum [GW95b] (en anglais Max-Cut *MC-u*) de taille u de sorte qu'il ait la même solution optimale. Ce problème consiste à partitionner les sommets d'un graphe d'arcs pondérés en deux ensembles, dont l'un est de taille u, de manière à ce que le poids total des arcs est maximisé. Le problème *MC-u* permet d'avoir une approximation fiable de *SDP* [GW95a, HYZ02]. Par conséquent, une solution approchée au problème de sélection d'un sous-ensemble de taille u peut être trouvé en résolvant *MC-u*.

Dans le cadre de la sélection à l'aide de test statistique, Tsoumakas et al. [TKV04, TAV05] proposent une méthode de sélection appelé *fusion sélective* qui combine les sorties d'un sous-ensemble de classifieurs sélectionnés à partir d'un ensemble hétérogène avec un vote pondéré. La sélection du sous-ensemble optimal est abordée comme un problème de comparaisons multiples, qui est résolu en appliquant des tests statistiques pour détecter des différences significatives des estimations des erreurs de prédiction dans la validation croisée. Dans [PTKV06], ces auteurs proposent d'utiliser l'apprentissage par renforcement afin d'identifier les sous-ensembles optimaux. Plus récemment, Meynet et Thiran [MT07] ont proposé une mesure théorique de la performance d'ensembles qui peut être utilisée pour la sélection d'un sous-ensemble à partir d'un ensemble initial de classifieurs.

De même que les algorithmes séquentiels, ces méthodes de sélection ne respecte pas le critère de complimentarité entre les membres sélectionnés de l'ensemble et que la sélection n'est pas conforme à la propriété des méthodes d'ensemble.

2.6.4 La sélection par classement

Les algorithmes de sélection par classement trient les modèles en fonction d'un certain critère et génèrent un ensemble contenant les m meilleurs modèles suivant le classement. La valeur de m est donnée ou déterminée sur la base d'un critère donné, à savoir un seuil, un minimum, ou un maximum.

Partridge et Yates classent les modèles en fonction de leur précision [PY96]. Ensuite, les m modèles les plus précis sont sélectionnés. Comme prévu, les résultats ne sont pas bons car la diversité de l'ensemble n'est pas garantie. Kotsiantis et Pintelas utilisent une

approche similaire [KP05]. Pour chaque modèle, un t-test est effectué pour comparer sa précision avec le modèle le plus précis. Les essais sont réalisés à l'aide d'une sélection aléatoire de 20% des données d'apprentissage. Si la p-value du t-test est inférieure à 5%, le modèle est rejeté. L'utilisation d'ensembles hétérogènes est la seule garantie de la diversité dans cette étude.

Pour tenter d'équilibrer la diversité et la précision, Rooney et al. les intègrent dans une mesure *score* [RPAT04]. En effet, en premier lieu, ils sélectionnent les modèles précis qui dépassent un seuil prédéfini. Chacun de ces modèles sélectionnés est caractérisé par une précision *Acc*.

$$Acc_i = \frac{E_{min}}{E_i^{sum}}, i = \{1, ..., n\}$$

où E_{min} est l'erreur minimale d'apprentissage de tous les modèles, et E_i^{sum} est la somme des erreurs d'apprentissage de tout les modèles sauf le $i^{ème}$.

Puis, à partir de l'ensemble formé de n modèles précis, ils calculent la corrélation entre ces membres. Les modèles ayant une corrélation dépassant un seuil prédéfini sont considérés comme fortement corrélés. Le nombre de fois m où la corrélation est estimée comme forte est calculé. Cette étape permet d'avoir une estimation de la diversité *Div* pour chaque modèle.

$$Div_i = \frac{n - m_i}{n}$$

Par la suite, un score de classement est calculé suivant la précision et la diversité de chaque modèle.

$$score_i = Acc_i + Div_i,\ i = \{1, ..., n\} \tag{2.3}$$

Finalement, seulement les k meilleurs sont sélectionnés (k valeur prédéfinie).

Liu et Yao [LY99] proposent d'utiliser la corrélation négative dans la recherche d'un sous-ensemble performant de réseaux de neurones. Les réseaux sont formés en utilisant une fonction de coût qui inclut un terme de pénalité de corrélation en plus de l'erreur de prédiction. Le terme de pénalité de corrélation encourage la spécialisation des différents réseaux et la coopération entre eux. Toutefois, la force de la pénalité doit être soigneusement réglée pour chaque problème. Si elle est trop petite, la coopération entre les membres de l'ensemble ne sera pas suffisante pour produire des améliorations significatives de performance. Si en revanche elle est trop grande, le processus d'apprentissage est inefficace. Autrement dit, la fonction de coût est dominée par le terme de pénalité et devient insensible aux erreurs de prédiction. Une autre difficulté de cette méthode est que la pénalité de corrélation introduit un couplage entre les paramètres des différents membres de l'ensemble. Ce couplage augmente la dimension de l'espace des paramètres dans lesquels la recherche est effectuée et rend l'apprentissage plus difficile. Dans la pratique, seuls les ensembles de petites tailles peuvent être construits avec cette technique.

D'autre part, Gacquer et al. [GDDP09] ont récemment proposé un algorithme génétique, appelé *DevGen*, qui permet un contrôle direct de l'importance accordée à la diversité et à la précision de l'ensemble des classifieurs (arbres de décision). *DevGen* réalise une

sélection d'arbres à partir d'une première série d'arbres de décision obtenus par boosting (en utilisant l'algorithme Adaboost). Cette sélection est basée sur la fonction de Fitness, initialement proposée par Optiz [MO11] pour la sélection des meilleurs sous ensembles de variables dans l'apprentissage d'un ensemble de classifieurs. Elle est calculée pour un ensemble de m d'arbres sur un ensemble de validation. La fonction de Fitness est définie comme suit :

$$Fitness(m) = \alpha \times Acc(m) + (1 - \alpha) \times Div(m) \tag{2.4}$$

- $Acc(m)$ est la valeur de précision des m arbres. Cette valeur est calculée sur un ensemble de validation à l'aide du vote majoritaire pondéré proposé pour l'algorithme Adaboost.
- $Div(m)$ est la valeur de la diversité de l'ensemble des m arbres. Elle est calculée par le test de $Kappa$ [C$^+$60]. Les arbres sont divers si la valeur de la mesure *kappa* est faible.
- α est le paramètre qui favorise l'importance soit de la précision soit de la diversité.

Comparé aux autres algorithmes utilisant un seul critère de performance pour la sélection (choose best ou Kappa Prunning) [PY96, MD97], DevGen donne généralement de meilleurs résultats. Ces résultats sont générés avec une valeur de α fixé à 0.8, c'est-à-dire, en donnant plus d'importance à la précision qu'à la diversité dans la sélection des modèles.

2.7 Application des méthodes d'ensemble dans l'inférence de réseaux de régulation

Basée sur les méthodes d'ensemble, une nouvelle famille d'algorithmes a récemment émergé dans le domaine d'inférence des réseaux de régulation, montrant un remarquable succès dans la prédiction des interactions inférées.

Parmi ces algorithmes, citons GENIE3 [HTIWG10], la méthode gagnante des challenges DREAM4 et DREAM5 [MSMF09, MSM$^+$10, MCK$^+$12]. GENIE3 est un algorithme basé sur un ensemble d'arbres de régression. Cet algorithme décompose la prédiction d'un réseau de régulation de p gènes en p sous problèmes de régression. Dans chacun de ces sous problèmes, un ensemble d'arbres de régression est construit pour prédire le profil d'expression d'un des p gènes (gène cible) à partir du profil d'expression de tous les autres gènes (gènes en entrée).

Spécifiquement, un ensemble de 1000 arbres est construit par gène. Chaque arbre parmi les 1000 construits produit un classement local des liens de régulation triés par importance. En effet, les arbres donnent la possibilité de calculer un poids d'importance qui permet de discriminer les gènes importants des gènes peu importants. Ceci détermine le poids des éventuelles interactions des gènes en entrée avec le gène cible en question. Par contre, l'utilisation triviale de ces poids peut introduire un biais positif dans les relations de régulation à l'égard des gènes fortement variables. Pour éviter ce biais, avant d'appliquer les méthodes d'ensemble à base d'arbres, les données d'expressions de gènes sont normalisées de façon qu'elles ont toutes la même variance sur l'ensemble d'apprentissage. Cette normalisation implique que les différents poids inférés à partir de différents modèles de prédiction sont comparables.

Une fois tous les ensembles d'arbres appris, le résultat final est la combinaison des différents classements locaux des p ensembles d'arbres en un classement global. Ce classement global représente la moyenne des poids pour chaque gène. Les étapes de GENIE3 sont illustrées dans la Figure 2.1.

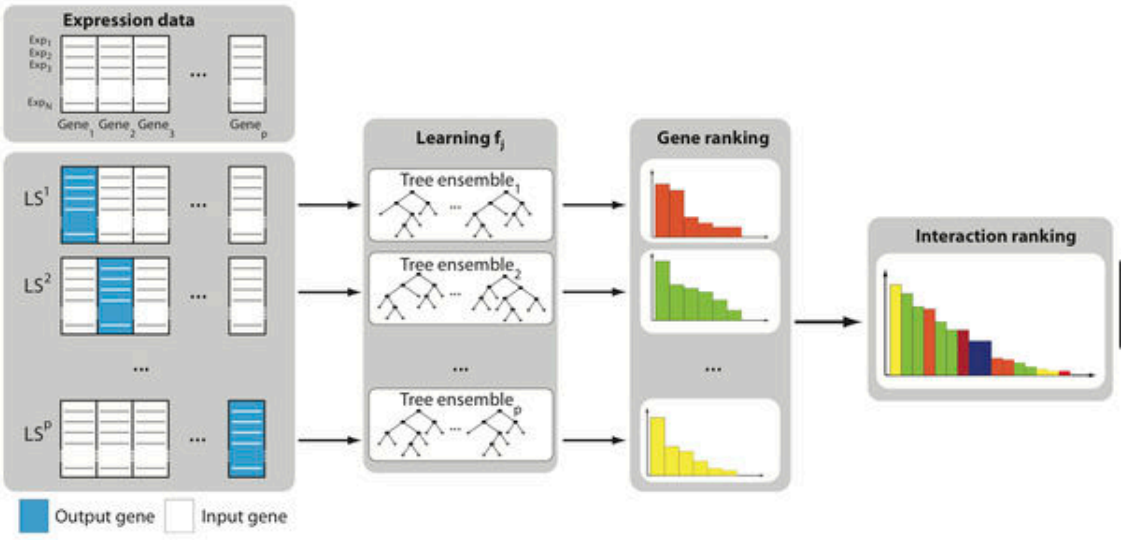


FIGURE 2.1 – Illustration graphique des étapes de GENIE3. Pour chaque gène $j = 1 \ldots p$, un sous échantillon d'apprentissage LS^j est généré avec les niveaux d'expression de j en sortie et les niveaux d'expression de tous les autres gènes en entrée. Une fonction (forêt aléatoire ou extra-arbre) f^j est apprise à partir de LS^j fournissant un classement local de tous les gènes (sauf j). Les classements locaux des p fonctions sont ensuite agrégés pour obtenir un classement global de tous les paires de régulation prédites. Figure extraite de [HTIWG10]

Deux types de construction d'ensemble d'arbres sont mis en place : les forêts aléatoires [Bre01] et les extra-arbres [GEW06]. L'expérimentation de ces deux techniques sur les données DREAM3 a prouvé qu'elles ont des résultats comparables. Toutefois, les challenges DREAM4 et DREAM5 ont été remportés par la construction d'ensemble sur la base des forêts aléatoires. Notons que sur les données DREAM4, GENIE3 arrive à bien classer les gènes qui ont un seul régulateur ou deux mais trouve des diffultés pour les gènes ayant plus de trois régulateurs. En d'autres termes, la qualité de classement de GENIE3 diminue si le nombre des régulateurs régulant les gènes augmente.

Une seconde approche est proposée par De Matos Simoes et al. afin d'améliorer la stabilité et la performance de l'algorithme *C3NET* [AES10, AES11], qui fait partie de la famille individuelle des méthodes d'inférence de réseaux de régulation (l'algorithme est présenté dans le chapitre précédent – voir Section 1.2.2).

L'idée principale de l'algorithme *BC3NET* [DMSES12] (pour bagging *C3NET*) est de générer un ensemble de B boostraps $\{D_k^b\}_{k=1}^B$ construits à partir des données

d'expressions $D(s)$ contenant s échantillons. Pour chaque boostrap D_k^b, un réseau G_k^b est inféré avec *C3NET*. A partir de l'ensemble des réseaux construits $\{G_k^b\}_{k=1}^B$, une

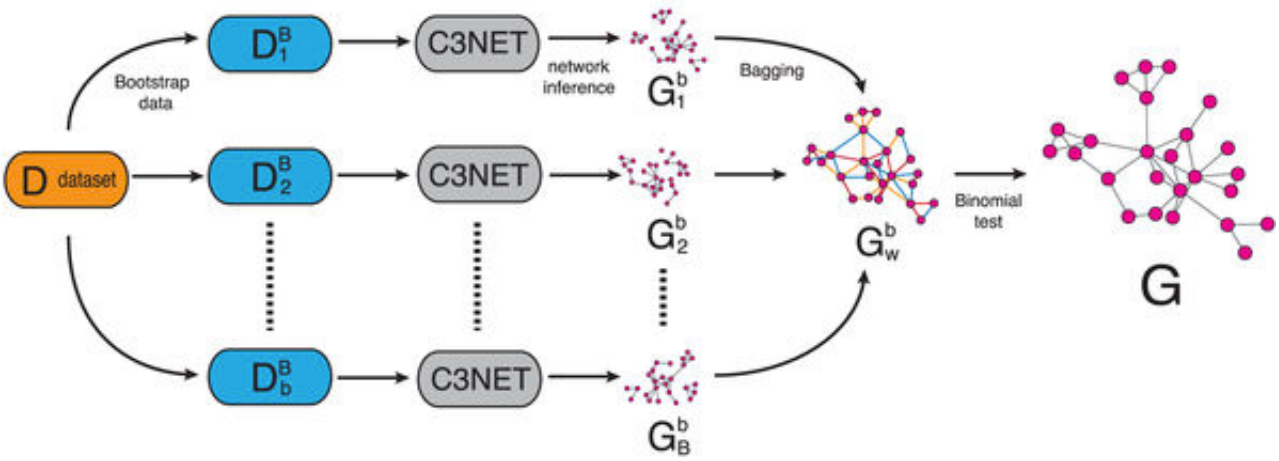


FIGURE 2.2 – Illustration graphique des étapes de *BC3NET*. Figure extraite de [DMSES12]

significativité statistique des relations entre les paires de gènes est estimée. Pour celà, les poids de répétition de toutes ces relations sont agrégés en un réseau G_w^b. Ensuite, seules les relations qui dépassent un seuil de signicativité sont considérées. Ce seuil est estimé par un test de distribution binomiale calculé en fonction de la taille des B boostraps et de la probabilité que les gènes peuvent être reliés par hasard. Le résultat de *BC3NET* est un réseau final G. La Figure 2.2 résume ce processus d'inférence. Les résultats expérimentaux sur des données *E.coli* montrent que *BC3NET* infère 40% de plus d'interactions correctes que *C3NET*.

Dans le même contexte, on trouve l'algorithme ADANET [SA12], qui s'est montré efficace dans l'inférence de réseaux. ADANET décompose le problème de l'inférence en un ensemble de problèmes de classification. Pour chaque ensemble, un seuil m est fixé pour la variable cible Y_k, $m \in [\gamma, \zeta]$. Le seuil m permet de définir le pourcentage d'échantillons qui seront considérés comme classe de "faible expression" (Ω_0) ou de "forte expression" (Ω_1) du gène k. Le seuil m est, par la suite, incrémenté progressivement jusqu'à atteindre la valeur limite ζ. Les étiquettes des classes sont définies par un vecteur binaire, où les échantillons qui appartiennent à la classe Ω_0 et Ω_1 sont étiquetés respectivement -1 et 1. Dans une deuxième étape, ADANET utilise l'algorithme de boosting Adaboost [FS+96] pour affecter un score à chaque régulateur potentiel en fonction de son pouvoir discriminant. En d'autres termes, il cherche les gènes qui pourraient discriminer des échantillons appartenant à la classe Ω_0 de ceux de la classe Ω_1. Enfin, il calcule la moyenne des scores d'importance sur tous les sous ensembles de classification. Les résultats expérimentaux d' ADANET montrent qu'il a de bonnes performances sur les données E.coli, et ce en comparant avec plusieurs algorithme de l'état de l'art, à savoir *C3NET*, CLR, ARACNE.

Récemment, une nouvelle méthode nommée TIGRESS [HMVLV12] a été évaluée la meilleure méthode de régression linéaire du challenge DREAM5. Le travail est une extension de l'utilisation de la régression LARS à l'inférence de réseaux [MGT09]. Comme expliqué

dans les méthodes d'inférence à base de régression linéaire (voir chapitre précédent–section 1.3.3), la régression LARS est très sensible et instable face à des variables explicatives corrélées (*i.e.*, partant des mêmes données, le modèle appris peut générer deux solutions différentes).
A l'aide des méthodes d'ensemble, TIGRESS surmonte cette limite. Dans le but de sélectionner des facteurs de transcription t ($t \in T_g$) qui prédisent l'état du gène g ($g \in G$), TIGRESS applique une procédure connue sous le nom de la sélection stable (en anglais stability selection)[MB10].

La sélection stable consiste à exécuter une méthode de sélection L fois avec une perturbation aléatoire des données et à estimer le score de chaque variable en fonction du nombre de fois où elle a été sélectionnée. Bach [Bac08] et Meinshausen et al. [MB10] ont démontré que la sélection stable réduit la sensibilité de la régression LARS et celle de la régression LASSO (qui ont toutes les deux le même problème d'instabilité). Ainsi, cette approche permet d'améliorer leur capacité à sélectionner les variables adéquates. Les scores des variables issues des multiples exécutions de la régression représentent la fréquence à laquelle chaque variable a été choisie dans ces exécutions.

La Figure 2.3 représente graphiquement cette fréquence pour un gène g fixé, et $L = 1, ..., 20$.

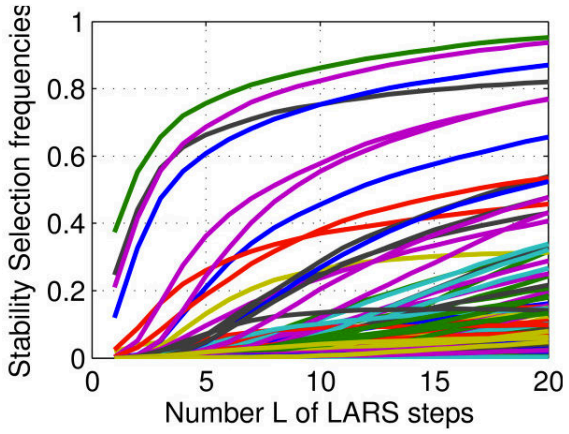


FIGURE 2.3 – Illustration de la fréquence de la sélection stable $F(g, t, L)$ pour un gène cible g fixé. Chaque courbe représente un facteur de transcription $t \in T_g$, l'axe des x représente le nombre L d'étapes LARS, et l'axe des y représente la fréquence $F(g, t, L)$ avec laquelle t est choisi dans les l premières étapes du LARS pour prédire g. Par exemple, le facteur de transcription correspondant à la courbe plus élevée a été sélectionné dans 57% des fois lors de la première étape du LARS, et 81% des fois après cinq étapes du LARS. Figure extraite de [HMVLV12]

2.8 Conclusion

Nous avons dressé dans ce chapitre une étude détaillée des méthodes d'ensemble ainsi que plusieurs algorithmes adoptant cette approche pour inférer de réseaux de régulation

Bien que ces méthodes aient prouvé leur efficacité à surpasser les contraintes inhérentes à l'inférence des réseaux de régulation, leur succès n'est que partiel lorsqu'elles sont

appliquées à des données biologiques réelles et ce même si les organismes sont relativement simples [WHRG03, WGH06].

L'explication principale de telles performances est que ces approches sont généralement globales. Ainsi, elles suggèrent un mécanisme de régulation relativement simple, dominé par un *seul* facteur de transcription, ce qui ne reflète pas réellement la complexité des mécanismes de régulation. En effet, il est communément admis que les régions régulatrices des gènes cibles peuvent contenir plusieurs sites de fixation pour plusieurs facteurs de transcription [LRR$^+$02, GF05, LJFJ06] et que ces facteurs de transcription interagissent souvent les uns avec les autres et forment des complexes de régulation pour agir (ensemble) sur leurs gènes cibles [HGL$^+$04, Cor09].

Dans cette thèse, nous tenons à respecter cette réalité biologique où plusieurs régulateurs agissent d'une manière coopérative pour influencer la transcription de leur gène cible. Ainsi, notre approche doit à la fois inférer des complexes de régulation tout en utilisant les méthodes d'ensemble afin de garantir de bonnes performances.

Avant de présenter plus en détail les contributions de cette thèse, nous exposons dans le chapitre qui suit une analyse critique de la méthode LICORN [ENBF$^+$07] (voir Section 1.2.3). Cette dernière a la particularité d'inférer des complexes de régulation étiquetés (*i.e.*, différentiation entre les activateurs et les inhibiteurs) que nous exploitons dans la création de notre algorithme.

Deuxième partie

Contributions

Chapitre 3

Analyse critique de Licorn

Sommaire

3.1 Introduction

Ce chapitre est consacré à l'analyse d'un algorithme d'inférence de réseaux locaux de régulation suivant une approche locale coopérative appelée Licorn (LearnIng CoOperative Regulation Networks) [ENBF+07]. Cette approche permet une inférence à large échelle et a la particularité d'inférer des complexes de régulation étiquetés (*i.e.*, différentiation entre les activateurs et les inhibiteurs). Néanmoins, nos tests ont révélé que Licorn présente des limitations, que nous exposons dans ce chapitre.

Ce chapitre fournit une évaluation détaillée de l'algorithme et met en évidence ses faiblesses. Il est organisé comme suit : la première section motive le choix de Licorn comme algorithme d'inférence. La Section 3.3 introduit le modèle de régulation coopérative adopté par cette approche et détaille l'algorithme d'inférence des réseaux locaux. Par la suite, on présente l'environnement de tests ainsi que les résultats obtenus, ce qui permet d'exposer les différentes limitations de Licorn. Finalement, la Section 3.6 propose les axes de recherches entrepris pour pallier à ces limitations et améliorer les performances.

3.2 Motivation

Le but de notre travail est de concevoir un algorithme d'inférence des réseaux de régulation performant et passant à l'échelle. Cette démarche doit aboutir à une méthode

qui répond à trois critères fondamentaux.

Premièrement, notre méthode doit être *performante :* elle doit pouvoir prédire un grand nombre de régulations réelles (*i.e.*, vrais positifs ou True Positive) tout en minimisant les régulations artifactes (*i.e.*, faux positifs ou False Positive). Pour celà, nous faisons recours au données DREAM afin de quantifier la performance de notre algorithme (voir Section B.3). Deuxièmement, nous devons respecter les *contraintes biologiques :* mettre en avant la coopérativité entre les facteurs de transcription qui agissent ensemble sur la régulation d'un gène cible en proposant un complexe *d'activateur* et un complexe *d'inhibiteur*. Finalement, notre algorithme doit avoir *de bonnes propriétés* (TP élevé, FP faible) même dans le cas de faibles échantillons.

Pour répondre à ces critères, nous avons choisi de bâtir notre solution autour d'un algorithme nommé LICORNdéveloppé au sein de notre équipe de recherche. Deux raisons principales motivent ce choix : (i) LICORN permet d'inférer des réseaux locaux coopératifs formés d'un complexe d'inhibiteurs et d'un complexe activateurs pour chaque gène et (ii) LICORN passe à l'échelle car il permet d'apprendre des réseaux complexes comme pour ceux des humains (*i.e.*, la complexité réside dans la taille du génome caractérisé par un grand nombre de gènes cibles, environ 30000, et un grand nombre de facteurs de transcription, environ 1700 [LJFJ06]).

Coopérativité Contrairement à la plupart des méthodes d'inférence (ARACNE[MNB+06], CLR[FHT+07], GENIE [HTIWG10], TIGRESS[HMVLV12], etc.) qui infèrent des relations par paire entre facteurs de transcription et gènes cibles, LICORN utilise un modèle local logique de régulation plus proche de la réalité. Il couvre les modes de coopération–concurrence opérant lorsque plusieurs régulateurs agissent sur un même gène cible [NKS05, CWC06]. Ceci est réalisé à l'aide d'un modèle de régulation logique, qui modélise la coopération entre facteurs (ET logique), et qui permet d'identifier le mode de coopération qui opère lorsque plusieurs régulateurs agissent sur un même gène cible. En d'autres termes, LICORN assure l'étiquetage *activation* ou *inhibition* des complexes de régulation agissant sur un gène cible. Ces relations étiquetées (activation/inhibition) ne nécessitent pas de post-traitement, à la différence des interactions apprises par des approches bayésiennes comme par exemple MINREG [FLNP00, PRT02]. LICORN a permis de découvrir des relations combinatoires entre régulateurs et leurs gènes cibles, ces résultats sont compatibles avec des résultats expérimentaux publiés (jeux de données de levure). De plus, cet algorithme est capable d'inférer des relations coopératives non identifiées par d'autres méthodes, comme par exemple les méthodes fondées sur des modèles bayésiens avec contraintes ou encore les arbres de décision [PRT02, SSR+03, MKW+04].

Passage à l'échelle L'inférence des réseaux de régulation à partir de données d'expression est un problème exponentiel en nombre de régulateurs (quelques centaines, voire des milliers chez les eucaryotes supérieurs). Pour contourner cette difficulté, et afin d'être capable d'apprendre des réseaux de régulation coopérative, LICORN considère la tâche d'inférence de réseaux de régulation comme un problème de recherche dans un espace d'états [Mit77], et ce, en optimisant le parcours de cet espace. LICORN met en œuvre une recherche adaptative pour traiter une contrainte d'optimisation [FKT00] qui sélectionne pour chaque gène cible les N-meilleurs co-régulateurs candidats. Cette stratégie réduit

considérablement le temps de calcul et fait face à la complexité engendrée par des données humaines (*i.e.*, le nombre important de régulateurs), tout en engendrant des réseaux de bonnes performances.

La suite de ce chapitre détaille l'algorithme LICORN. Dans un premier temps, nous présentons le modèle de régulation mis en place ainsi que la démarche adoptée pour construire les réseaux locaux de régulation. Ensuite, nous appliquons cette méthode sur des données DREAM5. Enfin, nous présentons les résultats et proposons les axes de recherches choisis afin d'améliorer les performances et pallier aux limites.

3.3 LICORN

3.3.1 Modèle local de régulation

Architecture du réseau de régulation

Le modèle de régulation de LICORN — de même que d'autres approches construisant des réseaux de régulation génétique (voir chapitre 1) — fait l'hypothèse suivante : les niveaux d'expression des régulateurs fournissent des informations au sujet de leur niveau d'activité de régulation. LICORN propose de représenter le réseau de régulation sous forme d'un graphe biparti :

- Le niveau $\mathcal{N}_1$ contient les régulateurs $\mathcal{R}$ (un nombre restreint estimé généralement à 10 % des gènes dans beaucoup d'organismes).
- Le niveau $\mathcal{N}_2$ contient les gènes cibles $\mathcal{G}$ (gènes sans activité de régulation).
- Les arcs $\mathcal{A} \subseteq (\mathcal{N}_1 * \mathcal{N}_1 * L)$ codent les relations de régulation entre les régulateurs et leurs gènes cibles, où chaque arc est étiqueté par un mode de régulation $L := \{activation, inhibition\}$ entre régulateur et gène cible.

Programme de régulation

Plusieurs méthodes simplifient le problème d'apprentissage initial et considèrent que les gènes peuvent être exprimés ou non *(on,off)* (voir Section 1.2). Ces approches ne peuvent pas capturer les niveaux intermédiaires de l'expression d'un gène, et peuvent facilement produire des résultats inexacts à cause de leur discrétisation binaire. Dans la formulation de LICORN, chaque gène, y compris les régulateurs, peut être dans l'un des trois états suivants : *normal, sur-exprimé ou sous-exprimé*. Cette discrétisation à trois niveaux représente mieux la réalité biologique qu'un modèle booléen, tout en permettant un passage à l'échelle. Plus précisément, l'activation (sur-expression) ou la répression (sous-expression) d'un gène cible g dans un contexte cellulaire particulier peut être expliquée par un effet transcriptionnel des activateurs de g, notés $A(g)$, ou des inhibiteurs de g, notés $I(g)$. Dans LICORN, une extension de la sémantique logique des modèles booléens [Kau69b, AMK99, LFS98] pour modéliser les mécanismes de régulation génétique a été utilisée. Les auteurs de [BGH03] ont démontré que des opérateurs logiques simples peuvent modéliser certaines régulations transcriptionnelles chez différents organismes. En particulier, certains gènes exigent que deux ou plusieurs régulateurs soient simultanément actifs (*i.e.*, sur-exprimés ou sous-exprimés) afin d'influencer la transcription de leur gène cible. Ces régulateurs forment des *complexes* d'activateurs ou d'inhibiteurs et fonctionnent selon la sémantique

de l'opérateur E_AND, vu comme une extension de l'opérateur logique *ET* à une logique à trois valeurs. Soit X un corégulateur (*i.e.*, tout sous-ensemble de régulateurs de $\mathcal{R}$), E_AND [1] est définit comme suit :

$$\text{E\_AND}(X) \;=\; \begin{cases} 1, & \text{si tous les } x_i \in X \text{ sont sur-exprimés ;} \\ \text{-1}, & \text{si tous les } x_i \in X \text{ sont sous-exprimés ;} \\ 0, & \text{sinon.} \end{cases} \tag{3.1}$$

L'état d'un gène cible g dans une condition biologique est une fonction discrète de l'état de ses régulateurs. Cette fonction, appelée *programme de régulation* (RP) calcule l'état estimé de g, appelé $\hat{g}_s(A, I)$, à partir des états combinés des activateurs A et des inhibiteurs I de g dans l'échantillon s (voir Figure 3.1).

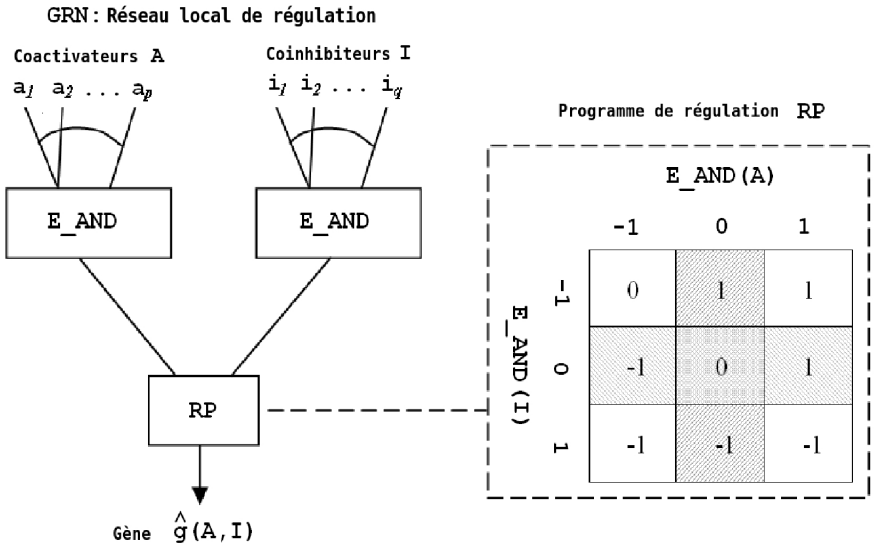


FIGURE 3.1 – (Figure extraite de [ENBF+07]) Représentation sous forme d'une matrice de décision du programme de régulation d'un gène cible g par un complexe d'activateurs A et d'inhibiteurs I. Cette matrice représente la valeur estimée de g, dénotée par $\hat{g}(A, I)$, en fonction de l'état de ses activateurs et de ses inhibiteurs. Ce programme de régulation est dissymétrique : si les activateurs et les inhibiteurs de g sont simultanément actifs alors le programme privilégie le rôle des inhibiteurs, d'où l'état de g qui est estimé à "sous-exprimé" (-1). Si A (resp. I) est vide, la matrice de décision se réduit à la colonne (resp. ligne) de la matrice correspondant à $\text{E\_AND}(A) = 0$ (resp. $\text{E\_AND}(I) = 0$).

Définition 1 GRN *On appelle GRN (de l'anglais Gene Regulatory Network) tout réseau local de régulation formé par un couple (A, I), de complexes d'activateurs A et de complexes d'inhibiteurs I, associé à un gène cible.*

1. $\text{E\_AND}(\emptyset) = 0$

Le *programme de régulation* RP (voir Figure 3.1) calcule l'état du gène cible selon les trois cas possibles suivants :

1. Si le GRN ne contient que des activateurs (*i.e.*,, $I = \emptyset$), RP spécifie la corrélation de l'expression estimée de g et de celle de ses régulateurs.
2. Si le GRN ne contient que des inhibiteurs (*i.e.*,, $A = \emptyset$), de façon duale, RP spécifie l'anti-corrélation de l'expression estimée de g et de celle de ses régulateurs.
3. Si le GRN contient à la fois des activateurs et des inhibiteurs, RP spécifie un *contrôle combinatoire* : par exemple, nous avons choisi qu'un gène cible est sur-exprimé lorsque ses activateurs sont sur-exprimés *et* ses inhibiteurs ne le sont pas, et il est sous-exprimé lorsque ses inhibiteurs sont sur-exprimés.

Les caractéristiques principales de ce modèle sont donc la représentation explicite des liens de régulation (activation/inhibition) et la modélisation de régulation coopérative.

3.3.2 Algorithme

LICORN a comme entrées les informations suivantes :
- un ensemble de régulateurs $\mathcal{R}$ et un ensemble de gènes cibles $\mathcal{G}$.
- deux matrices d'expression discrétisées (MR, MG) à valeurs dans $\varepsilon = \{-1, 0, 1\}$, codant respectivement les états de l'ensemble des régulateurs de $\mathcal{R}$ et des gènes cibles de $\mathcal{G}$ pour un échantillon $S = s_1, \ldots, s_n$ de n conditions
- un programme de régulation RP $: \varepsilon \times \varepsilon \to \varepsilon$ associe un état du gène cible à un état d'un ensemble d'activateurs $\subseteq \mathcal{R}$ et à un état d'un ensemble d'inhibiteurs $\subseteq \mathcal{R}$ (voir 3.3.1).
- une fonction de score local $h : \varepsilon^n \times \varepsilon^n \to \mathbb{R}$, permettant d'évaluer un réseau local de régulation.

L'algorithme de LICORN décompose la tâche d'apprentissage de structures pour chaque gène en trois étapes indépendantes : (i) l'extraction de co-régulateurs fréquents (ii) la génération des réseaux locaux de régulation (iii) l'évaluation de ces réseaux.

(i) Extraction de co-régulateurs fréquents

Chaque régulateur est représenté par deux ensembles supports,son 1-support $\mathcal{S}_1$ et son -1-support $\mathcal{S}_{-1}$. En utilisant une extension de l'algorithme *Apriori* [AIS93] – manipulant en parallèle les deux supports (1 et -1-supports)– LICORN construit le treillis CL des ensembles de co-régulateurs fréquents. Un co-régulateur est fréquent s'il est fréquent dans MR pour au moins une des valeurs d'intérêt, ici 1 ou -1.

Corégulateur fréquent *Étant donné une matrice d'expression à trois valeurs MR, un corégulateur $C \subseteq \mathcal{R}$ avec ses* 1- *et* -1*-supports, notés $\mathcal{S}_1(C), \mathcal{S}_{-1}(C)$, C est fréquent si et seulement si $max(|\mathcal{S}_1(C)|, |\mathcal{S}_{-1}(C)|) \geq S_{min}$, où S_{min} est un seuil minimum défini par l'utilisateur.*

Enfin, une valeur S_{min} faible est utilisée (*i.e.*, de l'ordre de 20 % de la taille de l'ensemble des échantillons), car cette étape vise à réduire l'ensemble des régulateurs candidats sans perdre les régulateurs peu exprimés dans le jeu de données.

(ii) Génération des réseaux locaux de régulation

Dans cette phase, l'espace de recherche est réduit à l'ensemble global de co-régulateurs fréquents (*i.e.*, le treillis CL). Pour chaque gène cible g, un ensemble de co-régulateurs candidats est calculé. Cet ensemble est caractérisé par une taille réduite par rapport à celle de tous les sous-ensembles possibles de régulateurs $2^{\mathcal{R}}$. Le critère retenu pour qu'un co-régulateur fréquent puisse participer au programme de régulation d'un gène cible est le fait d'observer dans les données d'expression une variation conjointe de leurs niveaux d'expression.

Dans la notion de *contrainte de co-régulation* entre un co-régulateur fréquent de CL et un gène cible g, chaque gène est représenté par deux supports : un 1-support $\mathcal{S}_1(g)$ et un -1-support $\mathcal{S}_{-1}(g)$ qui représentent respectivement les ensembles d'échantillons pour lesquels g est sur- et sous-exprimé. Il reste à vérifier l'intersection entre les supports du gène et ceux du co-régulateur à l'aide de la contrainte de co-régulation. Cette dernière est définie comme

Contrainte de corégulation *Soit un corégulateur C, un gène cible g, et leurs supports respectifs $\mathcal{S}_x(C)$ et $\mathcal{S}_y(g)$ pour les états $x, y \in \{-1, 1\}$. C co-régule le gène g, noté $C_{coreg}(\mathcal{S}_x(C), \mathcal{S}_y(g))$, si et seulement si $\frac{|\mathcal{S}_y(g) \cap \mathcal{S}_x(C)|}{|\mathcal{S}_y(g)|} \geq S_{coreg}$, où S_{coreg} est un seuil fixé par l'utilisateur.*

Cette étape de LICORN est contrôlée par cette définition de contrainte de co-régulation anti-monotone. La définition d'un seuil minimal pour la contrainte de corégulation, bien que très utile pour parcourir l'espace de recherche, est peu adaptée. En effet, les gènes peu exprimés vont avoir un nombre très élevé de co-régulateurs candidats non pertinents (*i.e.*, des co-régulateurs très fréquemment exprimés, mais néanmoins non corrélés avec l'expression du gène cible) entraînant un temps de calcul considérable.

En vue de passer à l'échelle, LICORN calcule de la manière la plus efficace les k meilleurs co-régulateurs de CL qui co-varient fréquemment avec le gène g. En fait, il ne s'agit plus de vérifier la contrainte de co-régulation isolément pour chaque régulateur (*i.e.*, recherche en largeur) mais de comparer les co-régulateurs entre eux et d'en extraire l'ensemble Sol des meilleurs candidats (*i.e.*, recherche du meilleur d'abord).

Recherche des k-meilleurs co-régulateurs Cette stratégie nécessite de fixer (i) le nombre k des solutions recherchées (*i.e.*,, la taille de l'ensemble Sol), (ii) S_{coreg} le seuil minimal sur la contrainte de co-régulation, (iii) un ensemble de co-régulateurs fréquents, dénoté $Ouvert$, trié par ordre de score de corégulation décroissant.

LICORN effectue alors une recherche de type meilleur d'abord dans le treillis des co-régulateurs CL. L'ensemble des noeuds en attente de développement est tout d'abord initialisé à l'ensemble des noeuds de niveau 1 de CL ($CL(1)$). Le noeud de meilleur score de $Ouvert$ ($Node$) est tout d'abord développé en recherchant ses successeurs dans CL. Les successeurs de $Node$ sont ajoutés à $Ouvert$ en fonction de leur score de co-régulation. Comme la contrainte de corégulation est anti-monotone, le score des successeurs de $Node$ est nécessairement inférieur ou égal au score de $Node$, assurant que tout nouveau noeud développé a un score inférieur ou égal à $Node$. Quand la taille de Sol atteint k, le processus d'ajout à Sol continue tant que le score du meilleur noeud de $Ouvert$ est le même que celui du dernier noeud de Sol.

Après l'identification des co-régulateurs pour chaque gène cible, une phase de distinction

selon leur mode de régulation (activateurs ou inhibiteurs) est mise en place. Les complexes candidats sont considérés activateurs, $\mathcal{A}(g)$, lorsqu'ils co-varient positivement avec le gène g, et inhibiteurs, $\mathcal{I}(g)$, lorsqu'ils co-varient négativement.

– Ensemble de complexes d'activateurs candidats :

$$\mathcal{A}(g) = \{A \in \text{CL} \mid C_{coreg}(\mathcal{S}_x(A), \mathcal{S}_x(g)); x \in \{-1, 1\}\}$$

– Ensemble de complexes d'inhibiteurs candidats :

$$\mathcal{I}(g) = \{I \in \text{CL} \mid C_{coreg}(\mathcal{S}_x(I), \mathcal{S}_{-x}(g)); x \in \{-1, 1\}\}$$

Ainsi, LICORN calcule l'ensemble de tous les réseaux de régulation candidats pour chaque gène cible g à partir des complexes d'activateurs et d'inhibiteurs extraits comme suit :

$$\mathcal{C}(g) = \{(A, I) | A \in \mathcal{A}(g), I \in \mathcal{I}(g) \text{ et } A \cap I = \emptyset\}$$

Un GRN candidat pour un gène cible g est un élément de $\mathcal{C}(g)$.

(iii) Évaluation des réseaux de régulation candidats

A ce niveau, un ensemble de réseaux candidats $\mathcal{C}(g)$ est construit pour chaque gène cible. A l'aide d'une fonction de score, un seul réseau local est sélectionné par LICORN. Cette fonction mesure à quel point un ensemble de régulateurs étiquetés (activateur, inhibiteur) permet d'estimer le niveau d'expression de son gène cible selon le programme de régulation RP.

Le score des GRNs candidats pour la régulation d'un gène donné est établi sur la base de la mesure des *moindres écarts absolus* (MAE, pour Mean Absolute Error) entre les profils d'expression observés du gène dans leur forme discrete et celui estimé par ses régulateurs candidats selon le programme de régulation *RP* (voir 3.3.1).

$$h_g(A, I) = \text{MAE}(g, \hat{g}(A, I)) = \sum_{s \in \mathcal{S}} |g_s - \hat{g}_s(A, I)| \tag{3.2}$$

où $g_s = \text{MG}_{sg}$ et $0 \leq \text{MAE} \leq 2$.

Enfin, le meilleur GRN pour le gène g est donc :

$$\text{GRN}^\star(g) = \underset{(A,I) \in \mathcal{C}(g)}{\text{Argmin}}\ h_g(A, I) \tag{3.3}$$

3.4 Cadre applicatif

Nous définissons dans cette section le cadre applicatif de l'ensemble des expérimentations que nous avons élaboré. Les expérimentations ont été effectuées sur les données du challenge DREAM5 [2] [MCK+12] préalablement discrétisées suivant la méthodologie présentée dans [ENBF+07].

2. http ://wiki.c2b2.columbia.edu/dream/index.php ?title=D5c4

3.4.1 Le challenge DREAM5

Nos expérimentations sont basées sur les donnés fournies dans le cadre du challenge DREAM5, l'un des plus récents jeux de données utilisé par la plus part des algorithmes [HTIWG10, HMVLV12, SA12] (voir Annexe B pour plus d'informations sur les challenges DREAM). Le challenge d'inférence DREAM5 évalue les algorithmes sur un réseau artificiel *in silico*. Ce jeu de données contient des niveaux d'expression de gènes provenant de différents types d'expériences, à savoir : les perturbations multifactorielles, la suppression de gènes ou encore des expériences sur-exprimées. Le réseau in silico est un ensemble de données simulées, générées en utilisant GNW [3] (version 3.0). Afin d'évaluer les résultats d'inférence, les interactions réelles (*i.e.*TP) entre gènes et régulateurs sont fournies (i.e. *grounds truth*) à travers la base Gold.

Le tableau 3.1 illustre le nombre d'échantillons, de gènes et de régulateurs (facteurs de transcription) présents dans la base in silico ainsi que le nombre d'interactions et la moyenne d'interactions par gène dans la base Gold.

	Dataset			**Gold dataset**	
	Échantillons	Gènes	Régulateurs	Interactions	#TP/gène $\mu(\sigma)$
In silico	805	1643	195	4012	$2.67_{(0.017)}$

TABLE 3.1 – Les caractéristiques de la base in silico du challenge DREAM5

3.4.2 La discrétisation des données DREAM5

LICORN adopte une discrétisation des données d'expression à trois valeurs (-1 : sous-exprimé, 0 : état normal et 1 : sur-exprimé) afin de préserver l'information biologique, contrairement à d'autres algorithmes discrets qui se basent sur une approche booléenne (*i.e.*, deux niveaux). La base DREAM5 est représentée par une matrice O de valeurs réelles d'expression de gène avec 805 échantillons ($|\mathcal{S}|$). La discrétisation de O en une matrice M à trois états (-1, 0 et 1) est réalisée comme suit :

Pour chaque entrée $\mathcal{S} = \{s_1, ..., s_n\}$ de n échantillons, soit $\mu_n(g)$ et $\sigma_n(g)$ la moyenne et l'écart type des niveaux d'expression réelles $O(g, s_i)$ de g dans $\mathcal{S}$ respectivement, et soit ρ est le paramètre qui règle la densité de la matrice résultante.

$$M(g, s_i) \quad = \quad \begin{cases} 1, & O(g, s_i) > (\mu_n(g) + \rho \cdot \sigma_n(g)) ; \\ \text{-1}, & O(g, s_i) < (\mu_n(g) \text{ - } \rho \cdot \sigma_n(g)) ; \\ 0, & \text{sinon.} \end{cases} \tag{3.4}$$

Nous avons choisi un seuil ρ conduisant à des fréquences équilibrées de -1 et 1 (*i.e.*, environ 15% de 1 et 15% de -1).

3. http ://tschaffter.ch/projects/gnw/

3.4.3 Le paramètrage de LICORN

Comme suggéré dans [CEN+], dans toutes nos expérimentations de LICORN nous fixons la taille de *Sol* et de *Ouvert* à 1% de la taille totale de l'espace de recherche CL. Ce dernier est de l'ordre de 22300 co-régulateurs fréquents obtenus pour un seuil de support minimum de 18%. Quand au seuil de corégulation S_{coreg}, nous le fixons à $0,2$ afin d'augmenter la probabilité de sélectionner les co-régulations rares.

3.5 Expérimentations et résultats

Cette section s'intéresse au réseaux locaux coopératifs générés par LICORN. Cette étude permet de mettre en évidence les faiblesses de cet algorithme.

3.5.1 Performances de LICORN

Dans cette première partie, nous évaluons la version initiale de LICORN qui infère un *seul* réseau local à chaque gène cible parmi un ensemble de candidats potentiels à la régulation de ce gène. Comme décrit dans la section 3.3.2, le classement des réseaux se fait suivant un score local MAE (voir équation 3.2) : un des réseaux ayant la plus faible MAE est sélectionné.

D'une manière générale, les méthodes d'inférence [MNB+06, HTIWG10, GH10, AES10, AES11, SA12, HMVLV12] évaluent la pertinence des interactions et non des réseaux. En effet, à chaque couple régulateur–gène cible, l'algorithme associe un score reflétant la pertinence de cette interaction. Cependant, LICORN quantifie la relation entre un gène cible et un complexe de régulation. Afin d'évaluer ce dernier, nous décomposons chaque GRN en l'ensemble des interactions qui le constituent. Chaque interaction a pour score celui attribué à son GRN. Dans le cas où une intéraction donnée est présente dans plusieurs réseaux locaux d'un gène cible, la moyenne des scores est calculée.

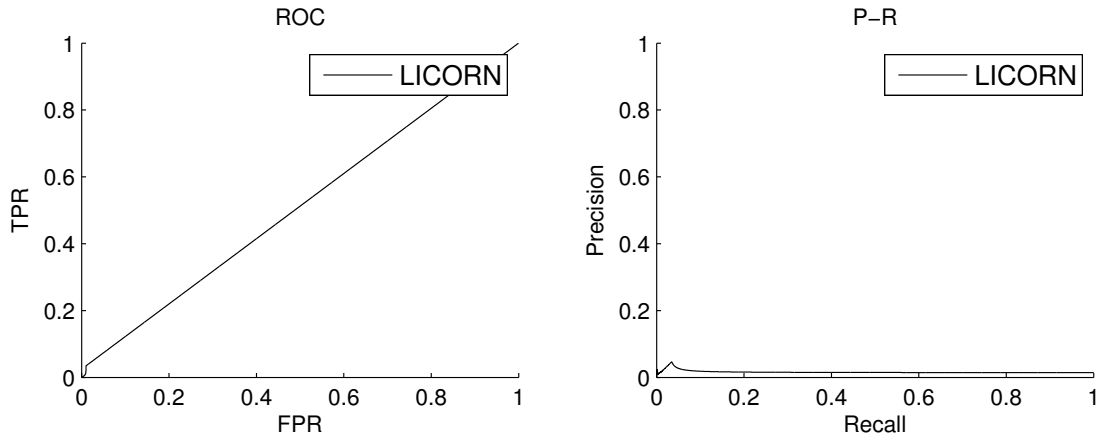


FIGURE 3.2 – Les courbes ROC et PR de LICORN sur les données DREAM5

La Figure 3.2 présente les courbes ROC et PR de LICORN. Les résultats obtenus indiquent que les GRNs inférés par LICORN ont de faibles performances. La courbe ROC

est presque linéaire indiquant un comportement proche d'une inférence aléatoire. En effet, le nombre d'interactions réelles (TP) inféré est très faible. Ces résultats sont également confirmés par la précision et le rappel de la courbe PR.

Ces résultats peuvent avoir deux explications : (i) LICORN n'infère pas des bons GRNs (*i.e.*, TP) (ii) LICORN est capable d'inférer les bons GRNs mais le score MAE ne permet pas de les trier correctement pour sélectionner le meilleur.

3.5.2 Performances des réseaux locaux candidats de LICORN

Nous faisons l'hypothèse que la deuxième explication est la bonne :(ii) nous mettons en doute la phase d'évaluation des GRNs candidats et la sélection du meilleur (voir étape 3 de LICORN 3.3.2). Afin de confirmer ou d'infirmer cette hypothèse, nous analysons plus en détails les GRNs candidats de LICORN. Ainsi, nous proposons d'étendre le nombre de candidats aux 100 GRNs ayant la plus faible valeur de MAE et que nous estimons suffisants pour avoir des 'bons' GRNs sans trop s'approcher des candidats sous-classés.

Par la suite, nous analysons la performance de l'algorithme sur ces 100 candidats (*i.e.*, au lieu d'un seul). Les résultats de cette approche nommée LICORN(100GRNs), ainsi que ceux de LICORN sont présentés par la Figure 3.3.

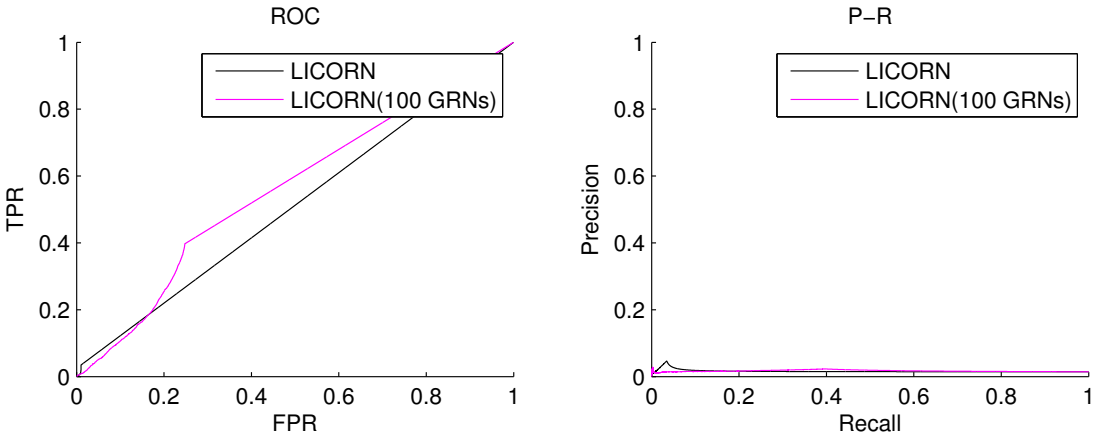


FIGURE 3.3 – Les courbes ROC et PR de LICORNet LICORN(100GRNs) sur les données DREAM5

La courbe ROC confirme notre hypothèse. En effet, nous remarquons un gain dans le taux des interactions TP (TPR) de LICORN(100GRNs) par rapport à celui de LICORN. Ceci indique que LICORN est capable d'inférer des bons GRNs (contenant des TP) mais le score MAE utilisé est incapable de les trier correctement. Il faut noter également que la courbe PR reste quasiment la même. En effet, garder 100 GRNs permet non seulement d'augmenter le nombre de TP, mais aussi le nombre de FP ce qui explique le faible rappel. En plus, la fonction permettant de classer les intéractions par leur importance reste inchangée par rapport à LICORN expliquant ainsi la faible précision.

Afin de quantifier les interactions réelles (TP) existantes dans les GRNs candidats et non prises en compte dans la phase de sélection, nous calculons les TP inférés par

Licorn et celles inférés par Licorn(100GRNs). Le résultat est présenté dans la Figure 3.4.

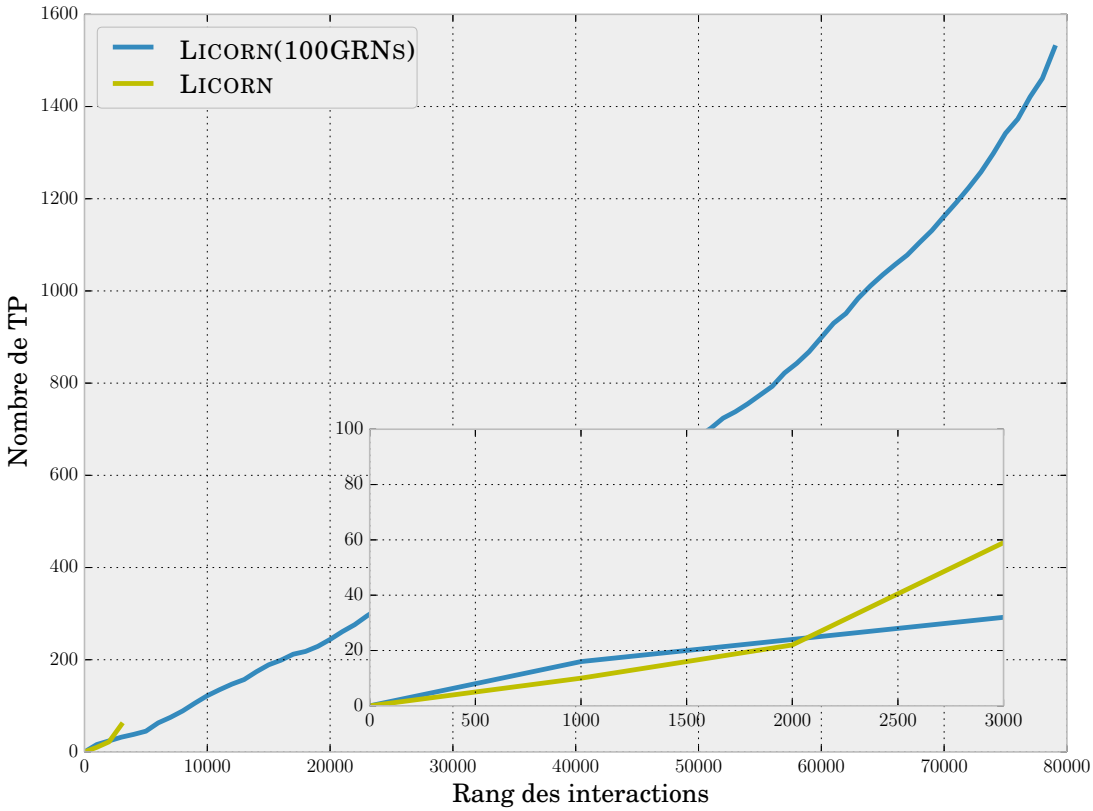


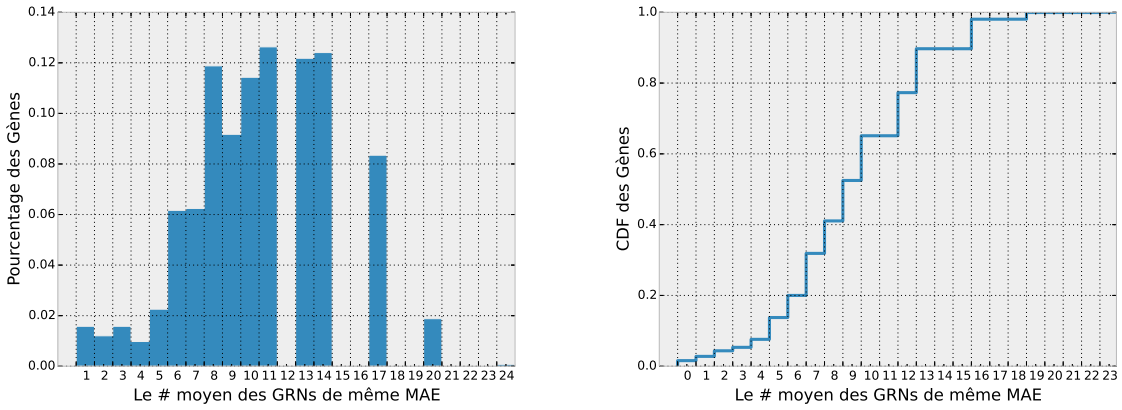
Figure 3.4 – Le nombre de TP inférés par Licorn(100GRNs) suivant le classement d'importance des interactions par le score MAE

Cette figure permet de faire deux constatations. Premièrement, un grand nombre d'interactions correctes TP sont inférées par Licorn(100GRNs) mais ne sont pas sélectionnées par Licorn(seulement 60 sur 1593). Deuxièmement, la courbe quasi linéaire de Licorn permet de déduire que le classement des gènes est quasi-aléatoire.

Pour conclure, les mauvaises performances de Licorn sont causées par deux processus : le classement et la sélection des GRNs. Dans ce qui suit, nous allons analyser le processus de sélection afin de proposer des améliorations.

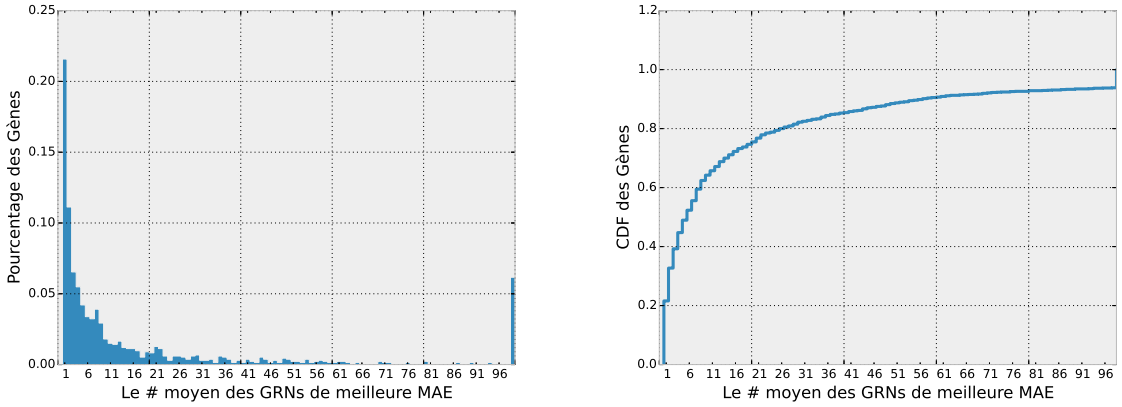
3.5.3 Performances de sélection

Les résultats précédents ont démontré que le choix d'un candidat en se basant uniquement sur la valeur de la MAE est insuffisant. Afin d'examiner la distribution de MAE pour chaque GRN candidat, nous effectuons une investigation sur la relation entre la valeur de la MAE et l'ensemble des GRNs candidats.



(a) (b)

FIGURE 3.5 – (a) Histogramme du nombre de gènes ayant en moyenne des GRNs candidats de même MAE (b) Fonction de répartition du nombre de gènes ayant en moyenne des GRNs candidats de même MAE



(a) (b)

FIGURE 3.6 – (a) Histogramme du nombre de gènes ayant en moyenne les GRNs de meilleure MAE (b) Fonction de répartition du nombre de gènes ayant en moyenne les GRNs de meilleure MAE

Premièrement, nous calculons le nombre moyen de GRNs candidats qui ont la même valeur de MAE pour un gène donné (Figure 3.5). L'histogramme de la Figure 3.5(a) démontre qu'une grande partie des GRNs candidats ont des scores égaux, d'où les pics qui sont atteints pour des valeurs supérieures à 5.La Figure 3.5(b) démontre que 50% des gènes ont plus de 10 GRNs candidats avec la même valeur de MAE et cette proportion atteint 80% des gènes si on considère 6 candidats ou plus.

Deuxièmement, nous calculons le nombre moyen de GRNs candidats qui ont la meilleure valeur de MAE pour un gène donné (Figure 3.6). La Figure 3.6(a) montre deux tendances pour le choix du meilleur GRN (*i.e.*, score MAE minimal) : (i) dans 35% des cas, le choix du meilleurs candidat se fait parmi un groupe de taille 2 : *i.e.*, il y a au maximum deux candidats ayant la meilleure valeur de MAE. (ii) Dans le reste des cas (*i.e.*, 65%), le choix est fait parmi un groupe de taille supérieure à 2 et pouvant atteindre 100. La longue traîne de la fonction de répartition (Figure 3.6(b)) montre bien cette dispersion de la meilleure valeur de MAE.

Ces constats permettent d'affirmer que la méthode de sélection est inadaptée vu que la valeur de la MAE n'est pas discriminante. En effet, étant donnés plusieurs candidats portant la même valeur de MAE, le choix du candidat se résume à un tirage aléatoire dans un ensemble de taille variable (de 1 à 100).

3.6 Discussion

L'étude expérimentale faite dans la section 3.5, nous a permis de mettre en évidence les forces et les faiblesses de Licorn. D'une part, Licorn génère des réseaux locaux coopératifs performants contenant un important nombre de TP, mais d'autre part, il n'arrive pas à sélectionner et à trier les bonnes interactions.

Sélection Le processus de sélection de Licorn est limité pour deux raisons. D'abord, le meilleur GRN est sélectionné parmi un ensemble de candidats portant la même valeur de MAE.Cette restriction à *un seul* réseau détériore sensiblement les performances (voir l'analyse de la section 3.5.2). Afin de palier à ce problème, nous proposons dans le Chapitre 4 un processus de sélection qui exploite les meilleurs réseaux inférés par Licorn (*i.e.*, les 100 premiers). Spécifiquement, le réseau "résultat" est construit à travers une combinaison de réseaux locaux qui maximise la capacité de la méthode à prédire des bon GRN. L'idée est de discriminer les GRNs porteurs d'interactions TP de ceux porteurs d'interactions FP. Pour ce faire, nous élaborons une méthodologie de sélection d'ensemble de GRNs performants à partir d'*un ensemble* de candidats. Nous basons ce processus sur la théorie des méthodes d'ensemble afin de garantir que les GRNs sélectionnés soient à la fois *précis* et *divers*.

Classement L'autre limite de Licorn réside dans l'évaluation des interactions obtenues. L'absence d'un processus de classement d'interaction détériore sensiblement les performances de l'algorithme. En effet, le classement de Licorn se base uniquement sur les GRNs sans prendre en considération les interactions qui les composent. Par conséquent, ce classement produit des résultats insatisfaisants démontrés par une très faible valeur de la précision. Le chapitre 4 propose plusieurs algorithmes afin de remédier à cette limitation. En particulier, nous proposons un consensus original de classement qui se base

principalement sur une technique numérique : la régression linéaire. Ce processus répond à deux pré-requis, à savoir, le respect de la structure des réseaux locaux (*i.e.*, complexe de régulation) et la maximisation de la fiabilité de chaque interaction.

3.7 Conclusion

Dans ce chapitre, nous avons motivé le choix de LICORN comme module basique pour la conception de notre algorithme d'inférence. Ce choix est motivé par deux raisons principales : la construction de réseaux de régulation coopératifs et le passage à l'échelle. A travers plusieurs expérimentations, nous avons démontré que LICORN a de sérieuses limitations qui impactent négativement ses performances. Spécifiquement, le processus de sélection et celui du classement sont peu performants. Le chapitre suivant propose plusieurs algorithmes afin de remédier à ces limitations.

Chapitre 4

Sélection d'ensemble de réseaux locaux de régulation

Sommaire

4.1 Introduction

Ce chapitre propose plusieurs solutions afin de remédier aux limitations de Licorn exposées précédemment (chapitre 3). Notre approche traite deux problématiques. La première consiste à substituer la phase de sélection peu performante par une autre basée sur les méthodes d'ensemble. Spécifiquement, on propose un algorithme glouton qui sélectionne un *ensemble* de GRNs suivant deux critères *i.e.*, la diversité et la précison. La seconde traite le classement des GRNs en proposant une méthode basée sur la régression linaire. Celle ci permet non seulement de quantifier la pertinence de chaque relation (*i.e.*, gène - régulateur) mais aussi la pertinence de chaque GRN. Ces deux modules sont intégrés au sein de l'algorithme SelectNet.

De plus, nous proposons une extension à SelectNet pour pallier à la faible *diversité* des GRNs inférés. Celle ci exploite la technique du *bagging* pour introduire de l'aléa et réduire la sensibilité au bruit.

Ce chapitre est organisé comme suit : La première section expose notre premier algorithme SelectNet. Tout d'abord, nous introduisons une méthode d'ensemble permettant la sélection de réseaux locaux suivant deux critères. Puis, nous présentons une approche numérique (*i.e.*, la régression linéaire) permettant un classement plus adéquat minimisant les faux positifs. Enfin, nous testons ces algorithmes sur les données du challenge Dream5 et nous le comparons au version initiale de Licorn. La section 4.3 traite l'introduction

de la diversité à travers le bagging. En premier lieu, nous motivons le choix du bagging comme moyen pour introduire de la diversité, puis, nous présentons SetNet la nouvelle version de SelectNet utilisant ce concept. Finalement, à travers plusieurs expériences nous comparons notre approche à deux algorithmes de l'état de l'art, Aracne et Genie3.

4.2 SelectNet

Dans le chapitre précédent, nous avons choisi d'utiliser Licorn comme module basique pour la conception de notre algorithme d'inférence. Néanmoins, nous avons démontré que Licorn présente de sérieuses limitations qui impactent négativement ses performances, en particulier au niveau des processus de sélection et de classement. Afin de remédier à ces limitations, nous proposons dans cette section une nouvelle approche d'inférence d'ensemble de réseaux de régulation coopératifs à partir de données d'expression, nommée SelectNet.

Pour chaque gène cible, SelectNet divise la tâche d'inférence en trois phases (voir l'Algorithme 2) : (i) l'extraction des GRNs candidats (ii) la sélection d'un ensemble de GRNs (iii) et le classement des GRNs ainsi que les interactions.

Extraction : SelectNet extrait pour chaque gène cible, un ensemble de E réseaux locaux candidats triés suivant une fonction locale de score. Cette phase exploite l'algorithme Licorn présenté dans le chapitre précédent.

Sélection : L'ensemble des E candidats est réduit à un sous-ensemble EG de N_s réseaux. Cette étape permet d'éliminer les GRNs non pertinents (*i.e.*, réduire les faux positifs) en s'assurant que les GRN de EG fournissent *collectivement* la meilleure prédiction de l'état du gène cible.

Classement : Finalement, nous trions les GRNs ainsi que leurs interactions en fonction de leur précision pour la tâche de prédiction de l'état du gène cible. Ce classement se base sur un modèle numérique de régression linéaire permettant de trier les interactions selon leur performance (*i.e.*, classer les vrais positifs en premier afin de maximiser la précision).

Algorithme 2 L'algorithme SelectNet

Entrées : Deux matrices numériques OR et OG respectivement pour les régulateurs et les gènes cibles

Sorties : Un ensemble de GRNs EG trié, pour chaque gène cible de $\mathcal{G}$; Un ensemble trié d'interactions (régulateur – gène)

1 : $EG := \emptyset$
2 : $MR := Discretiser(OR)$; $MG := Discretiser(OG)$;
3 : $E := Extraction\_De\_GRNs(MR, MG)$
4 : $EG := Selection\_De\_GRNs(MR, MG, E)$
5 : $Classement(EG, OR, OG)$

4.2.1 Extraction des réseaux candidats

A partir des données d'expression discrétisées MR et MG représentant respectivement les régulateurs et les gènes cibles, nous utilisons Licorn afin d'extraire un ensemble de

GRNs candidats pour chaque gène cible de $\mathcal{G}$. Notons que contrairement au fonctionnement par défaut de LICORN, où un seul GRN est choisi, nous gardons dans cette étape l'ensemble des meilleurs GRNs candidats pour chaque gène (noté E). Cet ensemble sera par la suite raffiné en vue de trouver un sous-ensemble de GRNs qui maximise les performances de prédiction de SELECTNET.

4.2.2 Sélection d'un ensemble de réseaux de régulation

Cet ensemble E doit être affiné afin de minimiser le nombre de faux positifs. Pour ce faire, nous allons extraire de E un sous-ensemble EG de N_s réseaux qui reflètent le plus fidèlement possible les données d'expression du gène cible g suivant le programme de régulation RP (voir Section 3.3.1).

Nous nous basons sur la méthodologie proposée par Dietterich [Die00a] pour la sélection d'ensembles qui explique qu'"une condition nécessaire et suffisante pour qu'un ensemble de classifieurs soit performant est que quelque soit le classifieur choisi, il doit être précis et différent des autres classifieurs". Ainsi, la clé de l'amélioration de la performance d'un ensemble est la complémentarité des prédictions fournies par les membres de cet ensemble. En effet, les erreurs de prédiction d'un membre peuvent être "compensées" par les bonnes prédictions des autres membres. Ainsi, l'utilisation séparée de la diversité ou de la precision n'améliore pas le rendement de l'ensemble [HS90] : seule une combinaison de ces deux critères peut le faire (voir Section 2.1).

Pour que l'ensemble EG soit performant, ces GRNs membres doivent à la fois être *précis* (*i.e.*, riches en interactions correctes TP) et *divers* (*i.e.*, contiennent des relations de régulation différentes). La méthode de construction de l'ensemble que nous utilisons est basée sur la méthode *Forward* proposée dans [RGV01]. Cet algorithme glouton n'ajoute le réseau à l'ensemble sélectionné que si ce réseau améliore au mieux le score global de l'ensemble. L'algorithme (voir Algorithme 3) est comme suit : étant donné un ensemble E de GRNs candidats à la régulation d'un gène cible g de $\mathcal{G}$, notés $net_{j(j\in[1..E])}$, l'algorithme commence par initialiser l'ensemble sélectionné avec le GRN le plus précis de E. Par la suite, et de manière itérative, le GRN candidat ayant la meilleure *fonction de score* est ajouté à l'ensemble sélectionné en cours de construction noté EG_c. Ce processus de sélection s'arrête lorsqu'aucun des GRNs candidats restants n'améliore la fonction de *Score* de l'ensemble EG_c. Nous proposons la fonction de *Score* suivante :

$$Score(net_j) = \theta \times Acc(net_j) + (1-\theta) \times Div(net_j) \tag{4.1}$$

La fonction *Score* peut être décomposée en deux parties :

- $Acc(net_j)$ représente la *précision* de la prédiction du complexe de régulation net_j par rapport à l'ensemble courant sélectionné EG_c. Ce score est calculé en utilisant le programme de régulation RP qui prédit un état $\hat{g}$ du gène cible g en fonction de l'ensemble des GRNs de EG_c (*i.e.*, précédemment inférés) et du complexe net_j en cours d'évaluation.

  $$\mathrm{RP}(EG_c \cup net_j) = \left\{\hat{g}_1, ..., \hat{g}_{|EG_c|}, \hat{g}_{net_j}\right\}$$

  Afin de calculer l'état du gène résultat $\hat{g}$, on utilise le vote majoritaire dans un espace discret $(-1, 0, 1)$. Chaque GRN de l'ensemble vote pour son état. L'état ayant

Algorithme 3 Sélection_De_GRNs par la fonction Score

Entrées : $E = \{net_1, \ldots, net_{|E|}\}$ l'ensemble de réseaux candidats inférés pour un gène donné ; $Score(net)$ la fonction de score

Sorties : EG, un ensemble de réseaux sélectionnés

1 : **Début**
2 : $EG = \emptyset$;
3 : $BN = select\_best\_score\_node(E)$; $E = E - BN$;
4 : **Tantque** $(R \neq \emptyset)$ AND $(Score \leq Score(EG_c \cup Node))$ **Faire**
5 : $Score := Score(EG_c \cup BN)$;
6 : $EG_c := EG_c \cup Node$;
7 : $BN = select\_best\_score\_node(E)$; $E = E - BN$;
8 : **Fin tantque**
9 : **Retourner** $EG = \{net_1, ..., net_{N_s}\}$
10 : **End**

le plus de votes est considéré celui du resultat $\hat{g}$.

$$\hat{g} = vote(\{\hat{g}_1, ..., \hat{g}_{|EG_c|}, \hat{g}_{net_j}\})$$

Finalement, la précision est calculée comme suit, en estimant l'erreur de prédiction entre l'état résultant $\hat{g}$ et l'état du gène cible g :

$$Acc(net_j) = 1 - Erreur(\hat{g}, g) \tag{4.2}$$

où

$$Erreur(\hat{g}, g) = \sum_{s \in \mathcal{S}} \mid g - \hat{g} \mid$$

– $Div(net_j)$ représente la *diversité* de net_j par rapport à l'ensemble de GRNs courant EG_c. Notons que nous considérons un réseau comme l'union de ses régulateurs sans prendre en compte le rôle des régulateurs (activateur ou inhibiteur). Deux approches sont proposées afin d'estimer cette valeur.
Tout d'abord, $Div(net_j)$ peut représenter la diversité prédictive de net_j par rapport à EG_c, que nous notons $Div\_pred(net_j)$. Dans ce cas, $Div\_pred(net_j)$ quantifie la différence entre la prédiction de l'ensemble des GRNs de EG_c obtenue lorsque net_j appartient à cet ensemble et celle obtenue dans le cas contraire. Concrètement, nous calculons la moyenne de la différence des prédictions entre net_j et les membres de EG_c à l'aide du RP.

$$Div\_pred(net_j) = \frac{1}{\mid EG_c \mid} \sum_{net_k \in EG_c} (\sum_{s \in \mathcal{S}} \mid \hat{g}_{net_k} - \hat{g}_{net_j} \mid) \tag{4.3}$$

avec

$$\hat{g}_{net_k \in EG_c} = \mathrm{RP}(net_k)$$

et

$$\hat{g}_{net_j} = \mathrm{RP}(net_j)$$

Alternativement, la structure des GRNs sélectionnés est aussi intéressante que le résultat de la prédiction. Nous définissons donc une mesure de diversité structurelle, que l'on nomme $Div\_stuct(net_j)$. Semblable à celle de la prédiction, cette mesure quantifie la différence structurelle entre l'ensemble courant (EG_c) et l'ensemble courant augmenté du candidat net_j. Nous cherchons à travers cette mesure à sélectionner des GRNs de structures différentes, qui sont par conséquent plus riches en interactions. Pour ce faire, nous utilisons l'*Indice de Jaccard* (IJ), rapide à calculer, qui permet d'évaluer la similarité entre deux ou plusieurs ensembles. Cet indice est défini par le rapport entre la cardinalité de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles. Pour n ensembles $(A_1, ..., A_n)$, on a :

$$IJ(A_1, ..., A_n) = \frac{A_1 \cap ... \cap A_n}{A_1 \cup ... \cup A_n}.$$

Dans notre cas, le but est de calculer la dissimilarité $Div\_stuct(net_j)$, que nous estimons en fonction de la similarité de IJ calculée entre net_j et tous les GRNs déjà sélectionnés dans EG_c.

$$Div\_stuct(net_j) = 1 - IJ(net_1, ..., net_{|EG_c|}, net_j) \tag{4.4}$$

avec

$$IJ(net_1, ..., net_{|EG_c|}, net_j) = \frac{net_1 \cap ... \cap net_{|EG_c|} \cap net_j}{net_1 \cup ... \cup net_{|EG_c|} \cup net_j}$$

Notons que la mesure de la diversité n'est possible que s'il y a au moins deux réseaux. Ainsi, cette valeur ne peut être calculée qu'à partir de la 2ème itération de l'algorithme. Ainsi, le choix du premier réseau de EG_c est basé uniquement sur la précision.

– Finalement, nous définissons le paramètre θ permettant de trouver un compromis entre la diversité et la précision. Ainsi :

$$\theta \quad \begin{cases} > 0.5, & \text{plus d'importance à la précision qu'à la diversité ;} \\ < 0.5, & \text{plus d'importance à la diversité qu'à la précision ;} \\ = 0.5, & \text{la même importance pour la précision et la diversité.} \end{cases} \tag{4.5}$$

A ce niveau de l'algorithme, la tâche de l'inférence d'un ensemble de réseaux locaux qui expriment au mieux l'état d'un gène cible est accomplie. Néanmoins, deux critères primordiaux pour améliorer la précision de notre algorithme d'inférence est encore manquant. En effet, nous devons fournir une méthode de tri des interactions régulateur–gène qui composent les GRNs : de la plus probable (*i.e.*, vrai positif) à la moins probable (*i.e.*, probablement faux positif). Cette tâche n'est pas simple car il n'y a aucune indication concernant l'importance d'une interaction au sein d'un GRN. De plus, nous devons impérativement respecter la structure des réseaux locaux (*i.e.*, les régulateurs composants chaque GRN). Dans ce qui suit, nous élaborons une approche qui obéit à ces deux critères.

4.2.3 Classement par double régression linéaire

Les algorithmes d'inférence [HMVLV12, HTIWG10, MNB+06] mettent en place une phase de classement afin de trier les interactions identifiées pour la régulation d'un gène.

Autrement dit, à chaque interaction prédite est associé un poids ou un rang exprimant l'importance et la fiabilité d'une telle relation. Cette tâche ne requiert pas d'algorithme spécifique si celui-ci infère des interactions simples (*i.e.*, interaction par paire régulateur – gène cible) [HMVLV12, HTIWG10, MNB$^+$06]. En effet, le classement n'est autre que le tri descendant des paires d'interaction en fonction de la pertinence.

Dans notre cas, le processus est plus complexe. En effet, notre approche infère des complexes de régulation. Ainsi, il faut impérativement pouvoir classer ces réseaux non seulement en fonction de la pertinence de chaque paire de relation, mais aussi en prenant en considération l'aspect coopératif (*i.e.*, réseau local). Afin de répondre à cette contrainte, nous utilisons la *régression linéaire*.

La régression linéaire est un modèle numérique de régression d'une variable expliquée sur une ou plusieurs variables explicatives. La fonction qui relie les variables explicatives à la variable expliquée est linéaire [Tib94]. Formellement, nous modélisons la relation entre une variable aléatoire expliquée y et un vecteur de variables aléatoires explicatives $(x_1, x_2, ..., x_m)$ comme suit :

$$y = \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_m + \mu \tag{4.6}$$

où μ désigne le terme d'erreur (parfois appelé perturbation). Nous supposons que nous disposons de données sur les variables $(y, x_1, x_2, ..., x_m)$, et nous cherchons à estimer le vecteur α des paramètres $(\alpha_1, \alpha_2, ..., \alpha_m)$.

D'un GRN à une régression linéaire

Le but est d'exprimer un gène cible g en fonction de ses régulateurs. Rappelons que la régression linéaire s'applique sur des données continues d'où l'utilisation de l'état numérique (*i.e.*, non discrétisé) du gène et des régulateurs (*i.e.*, respectivement OG et OR). Commençons par le scénario le plus simple où le gène g a un seul GRN composé par m régulateurs. La relation entre g et le GRN est alors transformée en une RL :

$$\hat{g} = \alpha_1 r_1 + \alpha_2 r_2 + ... + \alpha_m r_m + \mu \tag{4.7}$$

où $\hat{g}$ exprime une approximation de g et pour $i = 1, \ldots, m$, les r_i représentent les régulateurs du gène g pondérés par le poids α_i. Ainsi, classer ces interactions reviendrait à trier les régulateurs suivant leur poids : plus le poids est grand plus le régulateur associé est important.

Cependant, g peut avoir un ensemble de N_s GRNs ($N_s \geq 1$) ayant chacun un poids donné indiquant son importance dans l'ensemble. Notre approche doit agréger les résultats fournis par chaque GRN en un résultat final. Généralement, la stratégie d'agrégation utilisée est celle des moyennes [SAVdP08, AHVdP$^+$10, BS09, HGV11] : le résultat final est la moyenne des résultats obtenus par chacun des GRNs. Cette stratégie a un inconvénient majeur : l'importance des GRNs n'est pas prise en compte dans le résultat final, *i.e.*, tous les GRNs ont la même importance. Afin d'avoir un processus de classement plus précis, il faut impérativement combiner deux mesures : (i) une mesure *intra-réseau* pour quantifier le poids d'une interaction dans une GRN et (ii) une mesure *inter-réseaux* pour estimer le poids du GRN dans l'ensemble des GRNs appris (celui ci sert à classer les GRNs). Le poids final d'une interaction n'est autre que la somme des produits de son poids dans son propre

GRN et du poids de son GRN dans l'ensemble des GRNs sélectionnés. Ainsi, la régression est utilisée deux fois, d'où le nom de la double régression linéaire (voir algorithme 4)

Le classement intra-réseau est l'application d'une première RL au sein de chaque GRN parmi les N_s GRNs de EG.

$$\widehat{g}_{k(k\in[1..N_s])} = \alpha_{1k}r_{1k} + \alpha_{2k}r_{2k} + ... + \alpha_{mk}r_{mk} + \mu_k \tag{4.8}$$

Où $\widehat{g}_{k(k\in[1..N_s])}$ est la prédiction de l'état numérique de g à partir des états numériques de ses régulateurs $r_{i(i\in[1k...mk])}$.

Le classement inter-réseaux est l'application d'une deuxième RL au sein de l'ensemble des estimations résultantes $\widehat{g}_{k(k\in[1..N_s])}$ de l'étape précédente. Cette étape vise à évaluer le poids de chaque GRN par rapport à tous les GRNs présents dans EG. Ceci nous permet d'avoir un facteur de représentativité $\beta_{k(k\in[1..N_s])}$ d'un GRN k dans l'ensemble EG. La formule suivante représente la prédiction de l'état numérique de g, nommée $\widehat{G}$, à partir des états numériques des estimations $\widehat{g}_{k(k\in[1..N_s])}$:

$$\widehat{G} = \beta_1\widehat{g}_1 + \beta_2\widehat{g}_2 + ... + \beta_{N_s}\widehat{g}_{N_s} + \mu \tag{4.9}$$

Le rang final Le classement final d'une interaction (régulateur – gène cible), que l'on note $rang(r, g)$, est un compromis entre les deux mesures *intra-réseau* et *inter-réseaux*. Si une relation de régulation apparaît dans plusieurs réseaux, la somme des poids de ces relations est calculée comme suit :

$$rang(r_i, g) = \sum_{k=1}^{N_s(g)} \alpha_{ik} * \beta_k \tag{4.10}$$

Bien que la régression linéaire réponde à nos besoins, nous avons remarqué lors de la phase d'expérimentation une très grande variance entre les poids des GRNs au sein de l'ensemble (*i.e.*, β). En effet, dans certains cas, ces valeurs peuvent aller de 10^{-3} à 10^3. Afin de remédier à ce problème et de réduire cet écart dans l'estimation des poids, nous appliquons une généralisation de la RL, appelée régression ridge RR [HTF09]. Celle ci est conçue spécifiquement pour pallier aux grandes variations de ses coefficients en imposant une régularisation.

4.2.4 Expérimentations et résultats

Cette section est consacrée à l'étude des performances de l'algorithme SelectNet (algorithme 2). Nous utilisons le même cadre expérimental que celui présenté dans la Section 3.4 afin de pouvoir comparer notre approche à celle de Licorn.

Dans ce cadre, plusieurs paramètres sont à définir. Tout d'abord le paramètre θ de la fonction de score (voir équation 4.1) qui permet d'ajuster la diversité et la précision, puis, la taille de l'ensemble E des réseaux candidats extraits grâce à Licorn. Nous avons choisi de varier θ afin de quantifier son impact sur les performances. Par contre, la taille de

Algorithme 4 Classement par double régression linéaire

Entrées : Une matrice réelle O d'échantillons $\mathcal{S}$ pour les gènes et les régulateurs

Sorties : Un rang pour chaque GRN dans EG et un rang pour chaque interaction (régulateur – gène cible) de chaque GRN de EG

1 : **Pour tout** gène cible g de $\mathcal{G}$ **Faire**
2 : **Pour tout** GRN $net_k = (A_k, I_k) \in EG$ $(k \in 1..N_s(g))$ **Faire**
3 : $\widehat{g_k} = RR(A_k \cup I_k, g, O)$, avec les coefficients $(\alpha_{1k}, \ldots, \alpha_{mk})$.
4 : **Fin pour**
5 : $\widehat{G} = RR(\widehat{g_1}, \ldots, \widehat{g_{N_s(g)}}, g, O)$, avec les coefficients $(\beta_1, \ldots, \beta_{N_s(g)})$.
6 : **Pour tout** régulateurs r_i de g **Faire**
7 : $rank(r_i, g) = \sum_{k=1}^{N_s(g)} \alpha_{ik} * \beta_k$ avec α_{ik} le poids du régulateur r_i dans le réseau k (0 si r_i n'appartient pas au réseau k).
8 : **Fin pour**
9 : **Fin pour**

d'ensemble E a été fixée à 100. Nous suposons que ces 100 GRNs sont largement suffisants pour avoir un grand nombre d'interactions TP.

La première phase d'évaluation de SelectNet consiste à mesurer sa capacité à inférer les interactions réelles. Cette phase prend en considération deux aspects : l'extraction de vrais positifs (*i.e.*, interactions correctes) et le classement de ces interactions (*i.e.*, les plus probables en premier). Pour ce faire, nous utilisons les métriques usuelles, à savoir les courbes ROC et PR (voir Annexe B.2). De plus, nous comparons les GRNs obtenus aux interactions réelles fournies par la base *Gold*. Nous analysons, dans ce qui suit, les performances des deux étapes de l'algorithme : la sélection d'ensemble et le classement par double régression.

La sélection d'ensemble Une première analyse des courbes ROC et PR de la Figure 4.1 permet de mettre en évidence la supériorité de SelectNet par rapport à Licorn : les courbes ROC et PR de SelectNet (en couleur) sont systématiquement au dessus de celle de Licorn (en noir). Ces mauvaises performances de Licorn sont dues — comme démontré dans le chapitre précédent — à la restriction de l'inférence à un seul réseau candidat. L'approche ensembliste permet de résoudre ce problème. En générant 100 candidats, SelectNet est capable de fournir un grand nombre d'interactions réelles (TP). Ces interactions sont par la suite triées en utilisant la fonction score permettant ainsi de réduire sensiblement le nombre de fausses interactions (faux positifs ou FP). Ces résultats sont exprimés par une nette amélioration du TPR ainsi que du rappel (rappel entre 0.8 et 0.14).

La deuxième phase de notre analyse consiste à étudier l'impact du paramètre θ. La Figure 4.1 rapporte nos résultats. Tout d'abord, SelectNet dépasse Licorn quelque soit la valeur de θ. Deuxièmement, les résultats obtenus pour différentes valeurs de θ sont proches. Finalement, la sélection basée uniquement sur la précision *i.e.*, $\theta = 1$, est la plus performante. L'introduction de la diversité (structurelle Div_struc ou de prédiction Div_pred) dans le processus de sélection provoque une perte de performances par rapport à l'algorithme basé uniquement sur la précision.

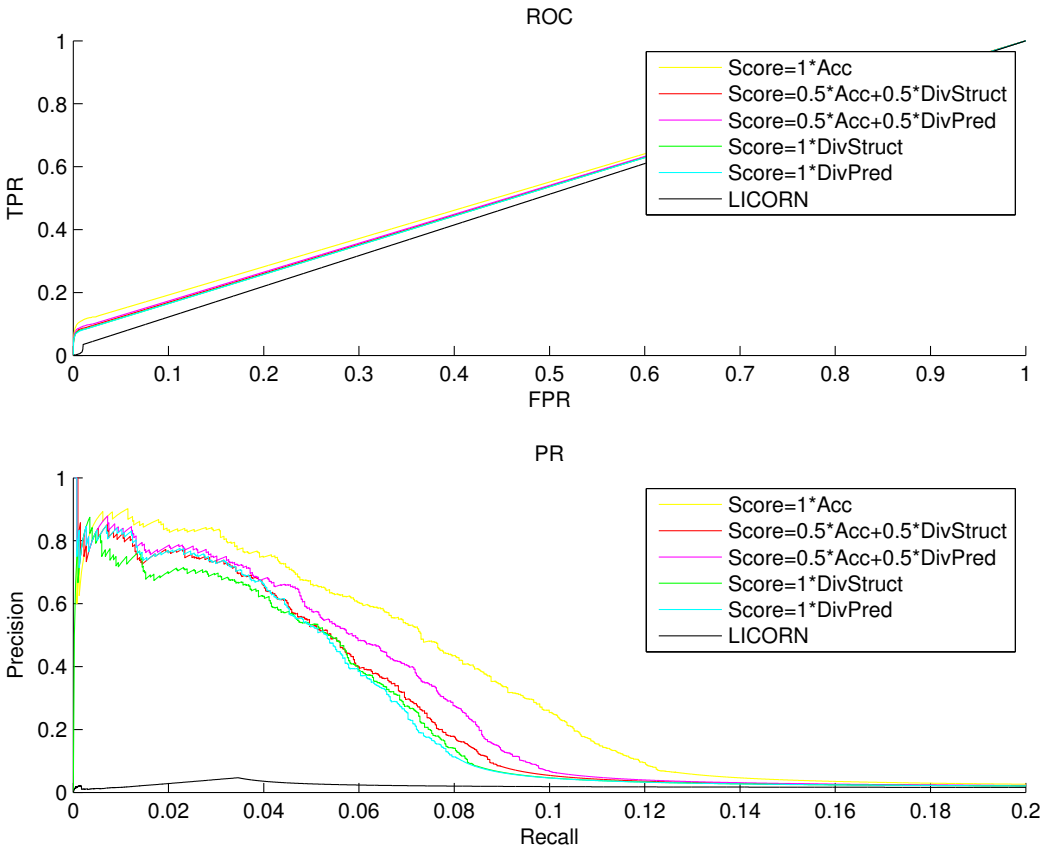


FIGURE 4.1 – Les courbes ROC et PR de l'algorithme LICORN et SELECTNET avec plusieurs variations de la fonction de score ($\theta \in (0, 0.5, 1)$) sur DREAM5

Le classement par double régression La courbe PR de la Figure 4.1 permet de comparer les performances du processus de classement des différents algorithmes proposés. En effet, la précision permet de mesurer la capacité de l'algorithme à trier les interactions suivant leur pertinence. Ces résultats démontrent que la double régression de SELECTNET est largement plus performante que le score MAE de LICORN (précision supérieure à 0.8 pour SELECTNET alors que LICORN est quasiment à zéro).

En plus, la Figure 4.2 présente le nombre d'interactions TP inférées par LICORN et celles inférées par SELECTNET (pour $\theta = 1$) selon leur classement par double régression. Nous constatons que le nombre de TP de SELECTNET est plus important que celui de LICORN. Dans les 1000 premières interactions, le nombre de TP passe de 20 à 360. La courbe montre également que la plupart des interactions TP de SELECTNET (environ 360 parmi 500) sont classées dans ces 1000 premières interactions, ce qui prouve que SELECTNET arrive à classer les interactions TP pour améliorer sa précision grâce à la double régression linéaire opérée sur les GRNs sélectionnés.

Ces résultats permettent de répondre à plusieurs faiblesses de LICORN et améliorent sensiblement les performances d'inférence. Néanmoins, ils soulèvent une question importante : *Quel est le rôle de la diversité ?* Nous rappelons tout d'abord que l'efficacité de l'ensemble repose à la fois sur la précision individuelle de ces membres ainsi que sur l'indépendance de leurs erreurs (*i.e.*, leur diversité) [Die00a, HS90, UN96]. Il est ainsi

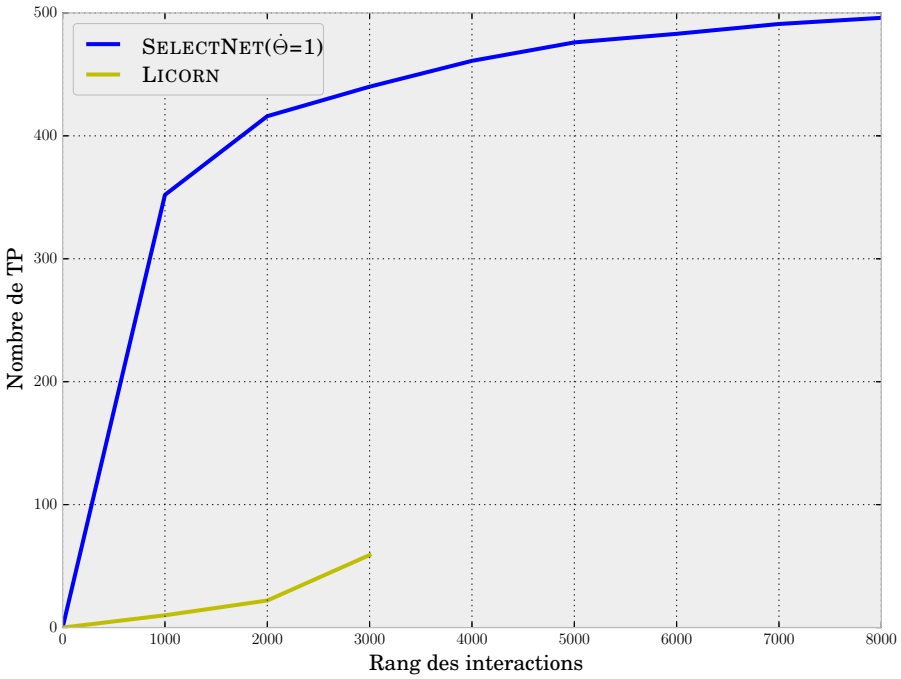


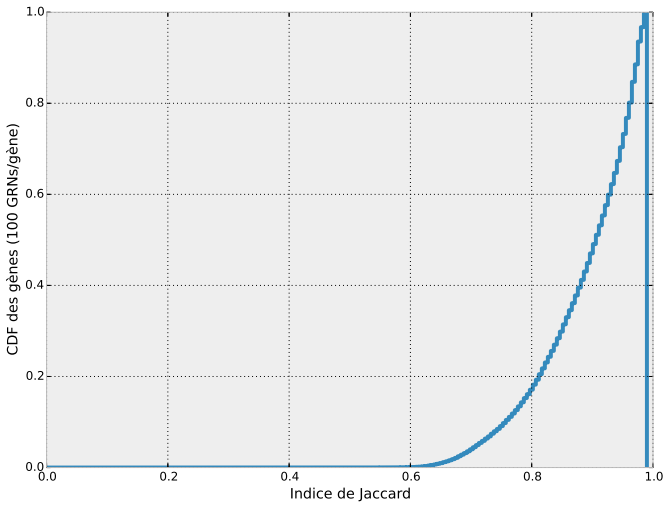
Figure 4.2 – Le nombre de TP inférés par Licorn et par SelectNet suivant le classement d'importance des interactions

connu que la combinaison de classifieurs similaires ne conduit pas à un gain de précision de prédiction de l'ensemble [UN96]. Parallèlement, la combinaison de classifieurs faibles peut conduire à une dégradation significative de la performance de l'ensemble [HS90]. Or, à travers nos expériences, nous avons constaté une dégradation des performances en introduisant la diversité, ce qui est paradoxal.

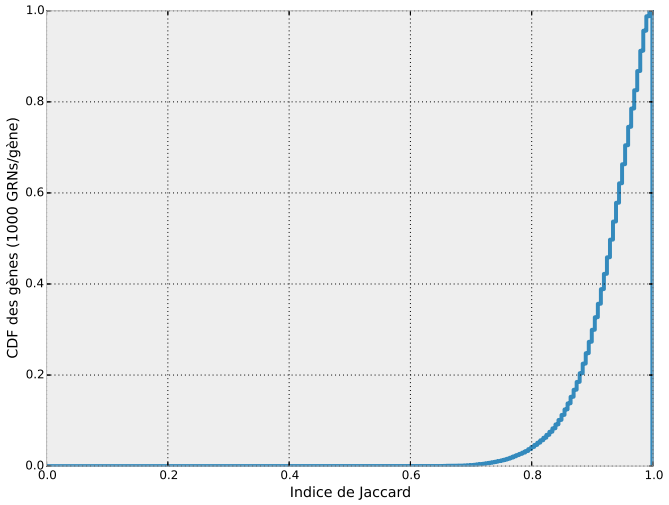
Pour mieux comprendre ce constat, nous mesurons la diversité entre les GRNs candidats. Nous cherchons à savoir *à quel point les GRNs candidats sont divers ?* A l'aide de l'indice de Jaccard, nous évaluons la diversité des 100 GRNs candidats pour chaque gène, ainsi que les 1000 GRNs candidats inférés par l'étape de l'extraction (*i.e.*, Licorn). La Figure 4.3 illustre les résultats.

La première fonction de répartition des 100 GRNs par gène (Figure 4.3(a)) montre qu'environ 80% des GRNs sont peu divers (indice de Jaccard supérieur à 0.8). Quant à la deuxième fonction de répartition (Figure 4.3(b)), elle montre que même avec l'extraction de 1000 GRNs candidats par gène, la diversité entre les GRNs n'augmente pas, au contraire elle génère des candidats encore plus similaires (95% des GRNs ont un indice de similarité supérieur à 0.8).

De ce fait, nous devons trouver une autre manière pour générer des GRNs candidats divers afin de respecter la définition de l'approche ensembliste et garantir ainsi l'amélioration des résultats. Pour celà, nous adoptons une approche pour introduire de la diversité tout en gardant la valeur de θ à 1. Notre approche consiste à perturber les données d'apprentissage avant l'inférence (*i.e.*, utilisation du bagging) [Bre96a, FS$^+$96, Ho98, MMS05]. Dans ce



(a)



(b)

FIGURE 4.3 – (a) Fonction de répartition de la moyenne de diversité de 100 GRNs candidats pour chaque gène (b) Fonction de répartition de la moyenne de diversité de 1000 GRNs candidats pour chaque gène

qui suit, nous présentons cette nouvelle méthode.

4.3 SetNet

Dans cette section, nous proposons une extension de l'algorithme SelectNet, que nous appelons SetNet. Cette extension est principalement basée sur la perturbation des données d'apprentissage.

Elle consiste à perturber les données d'apprentissage initiales en supprimant ou en ajoutant une partie de données ou encore en modifiant leurs poids relatifs dans l'ensemble d'apprentissage. La diversité entre les classifieurs dans l'ensemble est obtenue grâce à l'apprentissage de chacun d'eux sur une version perturbée différente des données initiales. Pour que cette stratégie soit efficace, les classifieurs individuels doivent présenter quelques instabilités. Cela signifie que des petits changements dans les données d'apprentissage devrait conduire à des changements dans les prédictions des modèles appris.

Plusieurs approches permettant de perturber des données d'apprentissage sont proposées, parmi lesquelles le Bootstrap [ET94]. Ce dernier construit un ensemble de données d'apprentissage en tirant aléatoirement des sous-ensembles des données réelles. Ainsi, les échantillons de bootstrap formés sont différents les uns des autres car ils partagent peu d'échantillons (voir Section 2.5.1). Le bagging [Bre96a] et les forêts aléatoires [Bre01] sont deux exemples de méthodes d'ensemble qui utilisent des échantillons bootstrap pour construire les différents classifieurs de l'ensemble.

Dans notre contexte, nous écartons l'utilisation des forêts aléatoires, car nous tenons à garder la structure des réseaux locaux coopératifs générés par Licorn. En effet, les forêts aléatoires sont formées par leurs propres classifieurs de base (*i.e.*,, arbre de décision ou arbre de régression). Nous optons donc pour le bagging qui a prouvé son efficacité [Bre96a, BK99, Die00b, Bre01] et son excellente performance dans de nombreuses tâches d'apprentissage d'intérêt pratique [HS90, KJS94, PC92, XKS92], entre autres dans les méthodes d'inférence de réseaux de régulations [DMSES12, HMVLV12, HTIWG10] (voir Section 2.7).

Dans notre cas, nous nous attendons à accroître la diversité entre les GRNs générés et à améliorer la précision de l'inférence en apprenant des GRNs sur plusieurs sous-ensembles de données tirés aléatoirement de l'ensemble initial d'apprentissage. De plus, le bagging joue un rôle important dans le processus de tri des GRNs et de leurs interactions. En effet, agréger les GRNs de tous les boostraps en un ensemble final de GRNs permet d'ajuster les poids de leurs importances ainsi que de leurs interactions. En d'autres termes, une interaction présente dans plusieurs GRNs séléctionnés à partir de différents boostraps renforce l'importance de ces GRNs et la fiabilité de cette interaction (*i.e.*, probablement c'est une interaction TP).

4.3.1 Algorithme d'inférence

L'algorithme d'apprentissage SetNet est exécuté sur un ensemble de N_b sous-ensembles de données tirés aléatoirement de l'ensemble d'apprentissage initial $\mathcal{S}$. Pour chaque sous ensemble $\mathcal{S}_{i(i\in[1..N_b])}$, deux étapes sont effectuées. Premièrement, l'étape d'extraction d'un ensemble $E_{i(i\in[1..N_b])}$ de GRNs candidats potentiels, pour chaque gène g

de $\mathcal{G}$, puis, l'étape de sélection d'un ensemble $EG_{i(i\in[1..N_b])}$ de GRNs parmi $E_{i(i\in[1..N_b])}$, suivant la fonction de score (voir équation 4.1). Une fois ces deux étapes achevées, les N_b sous-ensembles sélectionnés $EG_{i(i\in[1..N_b])}$ sont agrégés dans un ensemble global, noté EG_b. La dernière étape concerne le classement des GRNs, ainsi que leurs interactions et ce, avec le processus de double régression (voir Section 4.2.3). Ces différentes étapes sont illustrées dans l'algorithme 5.

Algorithme 5 SetNet

Entrées : Deux matrices numériques OR et OG respectivement pour les régulateurs et les gènes cibles

Sorties : Un ensemble de réseaux de régulation pour claque gène cible de $\mathcal{G}$

1 : $EG_b := \emptyset$
2 : $MR := Discretiser(OR)$; $MG := Discretiser(OG)$;
3 : **Pour tout** $i \in 1..N_b$ **Faire**
4 : Tirage aléatoire d'un sous-ensemble $\mathcal{S}_i$ de $\mathcal{S}$
5 : $MR_i := Projection(MR, \mathcal{S}_i)$
6 : $MG_i := Projection(MG, \mathcal{S}_i)$
7 : $E_i := Extraction\_De\_GRNs(MR_i, MG_i)$
8 : $EG_i := Selection\_De\_GRNs(MR_i, MG_i, E_i)$
9 : **Fin pour**
10 : $EG_b = \cup_{i\in[1..N_b]} EG_i$
11 : $Classement(EG_b, OR, OG)$

4.3.2 Expérimentations et résultats

Dans le contexte de l'évaluation de SetNet et afin de nous positionner par rapport à l'état de l'art, nous réalisons des comparaisons avec les algorithmes les plus performants en inférence de réseaux de régulation. En plus de Licorn, la méthode discrète utilisée pour l'extraction de réseaux locaux candidats dans sa version originale [ENBF$^+$07], et la méthode SelectNet, nous avons choisi de comparer notre approche avec deux algorithmes : Aracne [MNB$^+$06] et Genie3 [HTIWG10].

- Aracne [MNB$^+$06](voir Section 1.2.2) est une méthode discrète qui infère des interactions par paire (régulateur – gène) et (gène – gène). Sa matrice d'informations mutuelles MI est créée comme indiqué dans [MLB08]. Le seuil des interactions jugées fiables est fixé à la valeur par défaut (*i.e.*, 0).
- Genie3 [HTIWG10](voir Section 2.7) est une approche numérique basée sur les méthodes d'ensemble *i.e.*, les forêts aléatoires (voir Section 2.5.2). Comme Aracne, cette approche infère des interactions par paire (régulateur – gène) et (gène – gène). Son unique paramètre est le nombre d'arbres dans la forêt aléatoire, que nous avons fixé à 1000 comme suggéré dans [HTIWG10].

  Nous rappelons que Genie3 est la méthode gagnante du challenge Dream5.

Les paramètres de SetNet sont fixés comme suit : le nombre des sous-ensembles N_s à 100 afin de créer l'equilibre entre la durée d'exécution et une bonne couverture des GRNs. Nous optons pour une faible perturbation des données initiales. En effet, certaines études [KPJ$^+$03, SD02] ont démontré qu'une grande perturbation des données d'apprentissage

cause une instabilité du modèle. Ainsi, nous suivons les suggestions de [SAVdP08] en construisant SetNet sur des échantillons aléatoires contenant chacun 90% des données d'apprentissage. Rappelons que notre méthode infère des régulations de type (régulateur – gène). Afin d'établir une comparaison avec Aracne et Genie3 qui infère des relations (gène – gène) et (régulateur – gène), nous restreindrons l'inférence de toutes les méthodes au même type de relations (régulateur – gène).

Évaluation de performances

Sur l'ensemble des données Dream5, la Figure 4.4 et le tableau 4.1 résument les résultats obtenus par les différents algorithmes d'inférence.

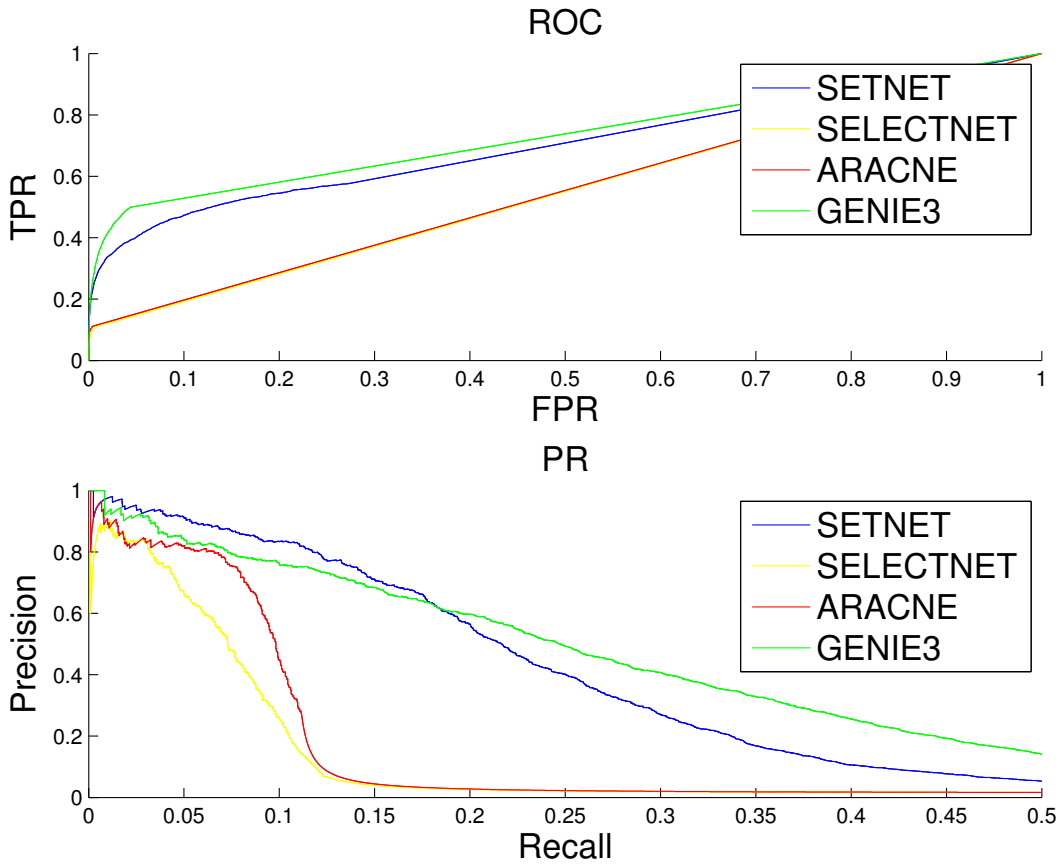


Figure 4.4 – Les courbes ROC et PR des différentes méthodes appliquées sur Dream5

Tout d'abord, nous remarquons une amélioration de l'algorithme générée par l'utilisation du bagging. Ainsi, SetNet surpasse tous les algorithmes sauf Genie3.

En effet, une forte amélioration est notée dans les valeurs AUROC et AUPR de SelectNet par rapport à celles de SetNet. Ceci est observé par les écarts entre les courbes jaunes de SelectNet et les courbes bleues de SetNet de la Figure 4.4 et par les valeurs de AUROC et AUPR du tableau 4.1 (respectivement, 0.67 pour SetNet face à 0.55 pour SelectNet et 0.18 pour SetNet face à 0.08 pour SelectNet).

Cette croissance de performances est expliquée par l'introduction de perturbations dans les échantillons d'apprentissage favorisant la découverte de nouvelles interactions.

Method	Dream5	
	AUROC	AUPR
SetNet	0.67	0.18
SelectNet	0.55	0.08
Aracne	0.55	0.10
Genie3	**0.73**	**0.26**

Table 4.1 – Les mesures AUROC et AUPR des différentes méthodes appliquées sur Dream5

Dans un deuxième lieu, la mise en comparaison met en valeur la performance de SetNet par rapport à Aracne et montre un léger écart par rapport à Genie3. Une explication possible des bons résultats de Genie3 est le fait qu'il utilise les forêts aléatoires qui font partie également des techniques d'ensemble (*i.e.*, le bagging d'arbre de regression).

D'après ces résultats, nous pouvons conclure que l'utilisation du bagging sur la base d'une méthode discrète (*i.e.*, SelectNet) fonctionne aussi bien que sur la base d'une méthode numérique (*e.g.*, arbre de regression).

Classement par double régression

La Figure 4.5 confirme nos résultats. En effet, l'introduction de perturbations aléatoires par la technique de boostraps a permis de découvrir encore plus d'interaction TP (environ 2300 par SetNet au lieu de 500 par SelectNet). En plus, l'agrégation de tous les GRNs résultants des différents tirages a permis d'affiner encore plus le classement des interactions. Par exemple, pour les 2000 premières interactions de régulation, SelectNet infère uniquement 400 TP, tandis que SetNet infère 700 TP.

Liens prédits

Afin de vérifier si les interactions réelles prédites (*i.e.*, faisant partie de la base Gold) sont les mêmes pour l'ensemble des algorithmes testés, nous comparons les intersections des différents ensembles de liens prédits par les trois approches (SetNet, Genie3 et Aracne).

Pour deux seuils de rappel (0.02 et 0.1), le nombre d'interactions inférées par les différentes méthodes est présenté par le diagramme de Venn de la Figure 4.6. Quant au choix des seuils, premièrement, nous optons pour une valeur de rappel de 0.02 où les trois méthodes ont une précision comparable. En effet, la précision maximale pour les trois méthodes est d'environ 0.9 pour cette valeur de rappel. Le deuxième seuil de 0.1 est fixé par rapport au rappel maximal d'Aracne. Au-delà de ce seuil, l'inférence de cet algorithme est aléatoire. Par conséquent, elle n'est pas pertinente dans le cadre de notre comparaison.

Une première constatation émanant de la Figure 4.6, est que les interactions TP et FP varient suivant la méthode indépendamment du rappel. Comme SetNet est conçu pour découvrir des interactions de régulation coopératives, nous nous attendons à ce qu'il infère des relations que d'autres approches ont des difficultés à découvrir. Ceci est confirmé par le plus grand nombre de TP propres à SetNet (*e.g.*, pour un seuil de 0.1, 191 TP sont

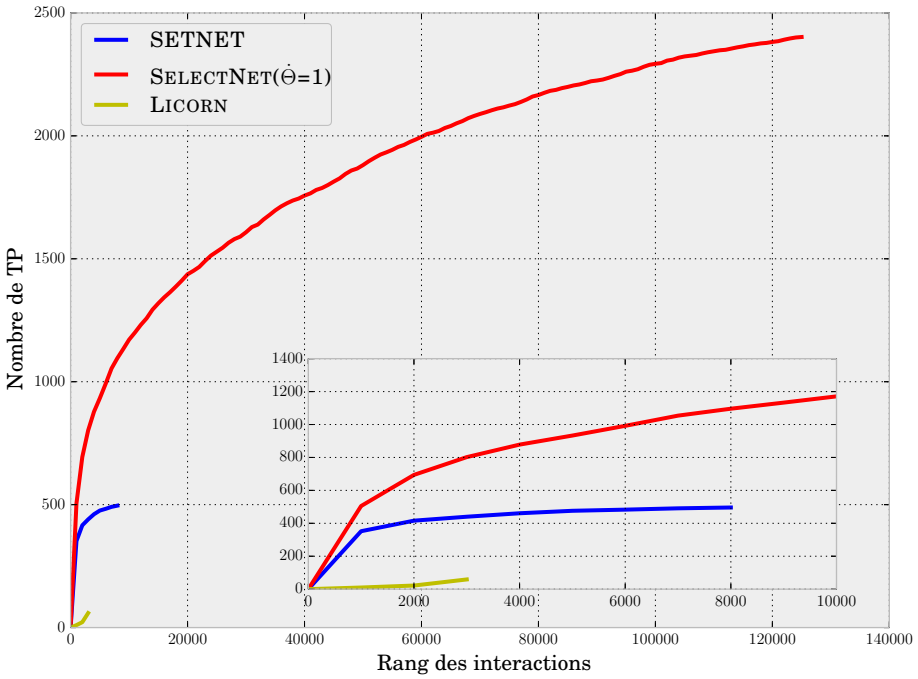


FIGURE 4.5 – Le nombre de TP inférées par LICORN , SELECTNET et SETNET suivant le classement d'importance des interactions

inférées uniquement par la méthode SETNET). Deuxièmement, le nombre total TP (en noir) inféré est quasiment le même pour les trois méthodes (81 pour un rappel 0.02 et 401 pour un rappel de 0.1). Néanmoins, une grande variation est observée dans le nombre de FP (en rouge). En effet, pour un seuil de 0.1, SETNET est la méthode qui infère le plus grand nombre de FP (236) alors que GENIE3 en infère moins (92 FP). Par conséquent, la capacité de SETNET à discriminer les TP des FP est mise en doute.

4.3.3 Discussion

La conception de SETNET vise à élaborer une méthode d'inférence performante. Nous avons analysé divers critères de performances à travers plusieurs expériences sur les données du challenge DREAM5. En dépit des mauvais résultats de LICORN initialement conçu pour prédire un seul GRN, SETNET a démontré une bonne capacité à prédire un ensemble de GRNs riche en interactions TP (voir Figure 4.4). Par ailleurs, nous avons intégré au processus d'inférence un processus de classement permettant de trier les interactions prédites selon un processus qui prend en compte le poids de l'interaction dans son propre GRN ainsi que le poids de son GRN dans l'ensemble. Ces poids permettent de donner plus d'importance aux interactions TP et par conséquent, diminuer le nombre de faux positifs.

Bien que performant sur le plan d'extraction de GRNs, SETNET pêche par le grand nombre de faux positifs. En effet, l'analyse des résultats de la Figure 4.6 montre que SETNET prédit des interactions TP qui ne sont pas identifiées par ARACNE et GENIE3 (*i.e.*, la phase d'extraction réussit) mais infère un grand nombre d'interactions FP (*i.e.*, la

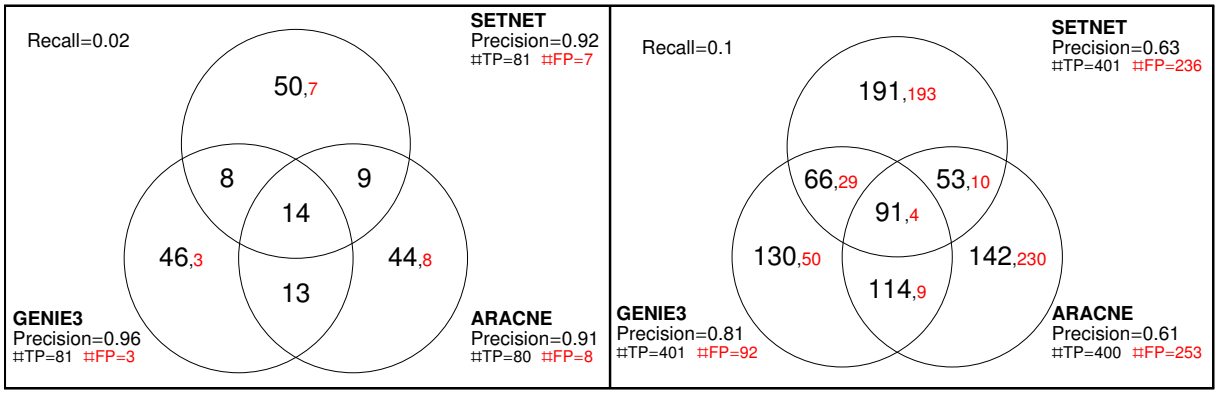


FIGURE 4.6 – Le nombre d'interactions TP et FP, respectivement en noir et en rouge, identifiées pour un rappel fixé (0.02 et 0.10) par SETNET, ARACNE et GENIE3 sur la base DREAM5

phase de sélection est problématique). Rappelons que ces deux phases d'extraction et de sélection dépendent fortement du programme de régulation discret RP (voir Section 3.3.1). Ainsi, la phase de sélection — contrairement a celle d'extraction — semble être sensible au données discrètes fournies par le programme de régulation générant ainsi un tri de GRNs non adéquat. Le chapitre suivant traitera cette problématique en proposant une méthode de sélection basée sur une approche numérique.

4.4 Conclusion

Dans ce chapitre, nous avons substitué la procédure de sélection ainsi que celle de classement de LICORN par deux nouvelles méthodes plus performantes. Pour cela, nous avons adopté une méthode ensembliste pour la sélection des données maximisant la performance et introduisant de la diversité. L'algorithme du classement des interactions a fait appel à une méthode numérique — la régression linéaire — qui a permis de prendre en considération aussi bien le poids d'un régulateur dans une interaction que le poids d'un GRN dans l'ensemble des GRNs inférés. Les diverses expérimentations menées sur les données DREAM5 ont démontré la supériorité de notre algorithme par rapport a LICORN. Dans un second temps, et afin de pallier au manque de diversité dans les GRNs prédits, nous avons proposé une extension exploitant la technique du bagging. Ce nouvel algorithme a démontré de bonnes performances en comparaison avec ARACNE et GENIE3. Néanmoins, il génère un nombre élevé de faux positifs dû une phase de sélection discrète peu performante. Le chapitre suivant proposera une nouvelle approche numérique performante pour sélectionner les GRNs et réduire ces faux positifs.

4.5 Bibliographie

Ce chapitre a été en partie publié dans :

– **Chebil, I**, Elati, M. and Rouveirol, C. and Santini, G. Hybrid method inference for the construction of cooperative regulatory network in human *Machine Learning and Applications (ICMLA), 12th International Conference on*, 2013.
– **Chebil, I**, Nicolle, R. and Rodrigues, C. and Rouveirol, C. and Elati, M. LICORN* : construction de réseaux de régulation chez l'homme *Conférence Francophone sur l'Apprentissage Automatique (CAp)*, 2012.

Chapitre 5

Méthode hybride pour l'inférence de réseaux de régulation

Sommaire

5.1 Introduction

Le chapitre précédent a présenté un nouvel algorithme d'inférence de réseaux locaux nommé SetNet permettant d'inférer des complexes de régulation à partir de données d'expression. Les diverses expérimentations menées sur la base Dream5 ont permis de faire deux constatations : notre approche infère un nombre important de vrais positifs et améliore ainsi les résultats de Licorn mais souffre d'un nombre élevé de faux positifs rendant SetNet moins performant que Genie3. Ce chapitre a pour but de proposer plusieurs solutions afin de pallier à ce problème. Rappelons que ce dernier réside dans la limite de SetNet à discriminer les bonnes interactions des mauvaises lors de la phase de sélection. Nous proposons une méthode *hybride* appelée H_SetNet qui génère —comme SetNet— des GRNs candidats suivant un modèle local discret de régulation (RP), et qui sélectionne, en seconde phase, les GRNs les plus précis —contrairement à SetNet— selon

une méthode de sélection *numérique*. Cette nouvelle stratégie de sélection abandonne le programme de régulation local discret peu précis pour une approche numérique. Dans la deuxième partie de ce chapitre, nous traitons le challenge d'inférence des GRNs dans le cadre de données humaines. Spécifiquement, nous évaluons les performances de notre méthode à identifier les régulateurs responsables du cancer de la vessie chez l'homme.

Ce chapitre est organisé comme suit : La Section 5.2 présente notre algorithme hybride H_SetNet. La Section 5.3 introduit deux techniques numériques de sélection (supervisée et non supervisée) susceptibles d'être intégrées à H_SetNet. Les deux sections suivantes (*i.e.*, 5.4 et 5.5) sont consacrées à l'étude expérimentale des deux techniques afin d'en choisir la meilleure. Une fois la méthodologie de sélection est fixée, H_SetNet est expérimenté dans la Section 5.6. Les résultats de H_SetNet sont présentés et comparés, dans un premier lieu, à SetNet. Ces résultats montrent une nette amélioration des performances par rapport à SetNet grâce à la baisse significative des faux positifs apportée par la sélection numérique. Dans un deuxième lieu, ces performances sont comparées à deux méthodes d'inférence *i.e.*, Aracne et Genie3 démontrant que H_SetNet présente des résultats meilleurs qu'Aracne et comparable à ceux de Genie3.

La seconde partie de ce chapitre traite la problématique d'inférence des GRNs humains. Comme ces données ont la spécificité d'être extrêmement sparses, nous effectuons une analyse de robustesse au sous échantillonnage dans la Section 5.7. A travers diverses expérimentations, nous montrons que notre algorithme est robuste. La dernière section 5.8 est consacrée à l'expérimentation de H_SetNet sur les données de cancer de la vessie. L'ensemble des résultats confirme que H_SetNet est en mesure d'identifier correctement des régulateurs importants de cancer.

5.2 H_SetNet

Nous avons présenté dans le Chapitre 4 l'algorithme SetNet d'inférence d'ensembles de GRNs coopératifs, pour chaque gène cible, à partir de données d'expression. Cet algorithme adopte une approche ensembliste reposant sur deux techniques : (i) la sélection d'un sous-ensemble de GRNs parmi plusieurs candidats à l'aide d'une fonction de score utilisant le programme de régulation discret RP (ii) l'exploitation du bagging afin de générer de GRNs divers et ainsi améliorer la précision globale.

Les expérimentations du chapitre précédent ont permis de faire deux observations. Le programme de régulation permet d'extraire des GRNs candidats riches en interactions positives et est donc indispensable lors de la phase d'extraction. Néanmoins, l'exploitation de ce même programme de régulation lors de la phase de sélection impacte négativement les performances de l'algorithme car il ne permet pas de trier convenablement les GRNs, ce qui engendre un grand nombre de faux positifs (voir Section 4.3.2).

Ainsi, il est primordial de substituer cette méthode de sélection peu précise par une procédure plus adéquate, indépendante du RP, permettant une meilleure discrimination des TP et FP. Nous proposons de changer l'étape de sélection de SetNet par une approche numérique donnant lieu à une méthode *hybride* nommée Hybrid SetNet ou H_SetNet (voir algorithme 6). L'originalité de notre approche réside dans sa capacité à exploiter le meilleur des deux espaces de recherche :

Espace discret pour la recherche et l'extraction de l'ensemble E de GRNs candidats

potentiels pour la régulation d'un gène cible de $\mathcal{G}$ (ligne 7 de l'algorithme 6). Cette phase utilise des données discrétisées (*i.e.*, MR et MG) révélant des propriétés importantes de réseaux locaux cooperatifs. En effet, elle est en mesure de prédire des interactions TP non identifiées par d'autres méthodes (voir l'étude des liens prédits dans 4.3.2).

Espace continu pour la recherche, dans l'ensemble des candidats E, du sous-ensemble EG des meilleurs GRNs suivant une technique numérique (ligne 8 de l'algorithme 6). Cette phase utilise des données continues (*i.e.*, OR et OG) modélisant explicitement les niveaux d'expressions et permettant des modèles plus précis des réseaux candidats. En effet, dans cet espace, toute l'information sur les niveaux d'expression est exploitée (*i.e.*, pas de perte d'information suite à la discrétisation).

Le choix d'une telle méthode doit impérativement répondre à deux critères : (i) une amélioration des capacités d'inférence à travers la réduction des faux positifs et (ii) la préservation de la structure des GRNs inférés.

Algorithme 6 H_SetNet

Entrées : Deux matrices numériques OR et OG respectivement pour les régulateurs et les gènes cibles

Sorties : Un ensemble de réseaux de régulation pour chaque gène cible de $\mathcal{G}$

1 : $EG_b := \emptyset$
2 : $MR := Discretiser(OR)$; $MG := Discretiser(OG)$;
3 : **Pour tout** $i \in 1..N_b$ **Faire**
4 : Tirage aléatoire d'un sous-ensemble d'échantillons $\mathcal{S}_i$ de l'ensemble d'échantillons $\mathcal{S}$
5 : $MR_i := Projection(MR, \mathcal{S}_i)$
6 : $MG_i := Projection(MG, \mathcal{S}_i)$
7 : $E_i := Extraction\_De\_GRNs(MR_i, MG_i)$
8 : $EG_i := Selection\_Numérique\_De\_GRNs(OR_i, OG_i, E_i)$
9 : **Fin pour**
10 : $EG_b = \cup_{i \in [1..N_b]} EG_i$
11 : $Classement(EG_b, OR, OG)$

5.3 Sélection numérique

Plusieurs approches numériques peuvent répondre aux critères cités précédemment. Dans ce qui suit, nous étudions deux d'entre elles :

1. **Un modèle numérique non supervisé : le clustering.** C'est le regroupement des GRNs candidats en clusters afin d'identifier les GRNs les plus représentatifs (*i.e.*, *représentants* du cluster) pour chaque gène cible de $\mathcal{G}$.
2. **Un modèle numérique supervisé : la régression linéaire.** C'est la modélisation de chaque GRN par une régression linéaire permettant d'attribuer un *score* à chaque candidat. Ce score correspond à l'erreur de prédiction de l'état du gène cible. Les GRNs les plus précis sont par la suite sélectionnés.

5.3.1 Partitionnement des GRNs

Le partitionnement de données (ou clustering en anglais) vise à organiser les données en différents "paquets" (ou clusters) homogènes. Cette homogénéité exprime une ressemblance entre les données appartenant à un même cluster et correspond le plus souvent à des critères de proximité ou de similarité exprimés en fonction d'une mesure de distance (*e.g.*, distance de Jaccard, distance euclidienne, etc). Cette technique est utilisée dans divers scénarios [BH03, GR01, GRF00, LO01, CVZ06] afin d'extraire *le* ou *les* éléments les plus représentatifs de chaque cluster. Certaines de ces techniques supposent que le nombre de clusters est un paramètre fourni par l'utilisateur (*e.g.*, k-means, k-medoids) contrairement à d'autres (*e.g.*, DBSCAN [EKSX96] et OPTICS [ABpKS99]) où ce nombre résulte du partitionnement.

Afin de sélectionner les GRNs candidats les plus représentatifs, nous utilisons le clustering comme suit : l'espace E de tous les GRNs candidats pour un gène g est partitionné en un ensemble de clusters contenant chacun un sous-ensemble de GRNs "similaires". On sélectionne par la suite *le* représentant de chaque cluster réduisant ainsi l'ensemble de GRNs inférés à ceux qui sont les plus représentatifs. Nous utilisons une mesure de similarité basée sur la *similarité structurelle* en regroupant les GRNs qui sont structurellement proches (*i.e.*, ayant des complexes activateurs/inhibiteurs semblables) dans un même cluster. Pour ce faire, nous adoptons les *cartes auto-organisatrices* [Koh95].

Ce choix est motivé par les nombreux avantages de cette approche. Tout d'abord, la prise en considération de la *topologie* des données permet d'avoir des clusters qui prennent en considération la notion de voisinage. Ensuite, comme le nombre de clusters est inconnu, une approche à la k-mean suppose non seulement une connaissance de ce paramétre (k) [1] mais en plus ne permet pas *dans la majorité des implémentations* d'avoir des clusters vides. Les cartes auto-organisatrices permettent de pallier à ce problème en fournissant un ensemble de cluster cohérents, noté N_s ($N_s \leq k$), en générant des cluster vides si le paramètre k est trop élevé. Finalement, la faible complexité de cet algorithme permet une expérimentation sur des grandes données.

Les cartes auto-organisatrices

Les cartes auto-organisatrices (en anglais *Self-Organizing Maps SOM*), ont été proposées par Kohonen [Koh95]. Elles rentrent dans la catégorie des méthodes de classification par partitionnement. C'est un type de réseau de neurones artificiels qui projettent des données multidimensionnelles sur un espace de faible dimension, souvent en 2D, dans le but d'effectuer des tâches de discrétisation ou de classification. Une carte auto-organisatrice bidimensionnelle est composée de cellules disposées sur une grille rectangulaire ou hexagonale. La Figure 5.1 représente une carte auto-organisatrice carrée contenant 49 cellules. À chaque cellule est associé un représentant de cet espace de données, nommé *vecteur référent*, et un emplacement sur la carte (*i.e.*, ligne et colonne de la cellule). Plus les cellules sont proches, plus la similarité est importante. Ceci est illustré par la cellule 25 (en rouge) contenant des données relativement similaires aux cellules voisines (lien vert).

La carte se présente sous forme d'une grille possédant un ordre topologique de K

1. ou du moins suppose un pré-traitement pour estimer cette information

cellules. Les cellules sont réparties sur les nœuds d'un maillage. La prise en compte dans la carte de la notion de proximité impose de définir une relation de voisinage topologique. L'influence mutuelle entre deux cellules c_1 et c_2 est donc définie par la fonction $\mathcal{K}^{\mathcal{T}}(\delta(c_1, c_2))$ où $\delta(c_1, c_2)$ est la distance entre les deux cellules c_1 et c_2. Chaque cellule c de la grille $\mathcal{C}$ est associée à un vecteur référent $\mathbf{w}_c = (\mathbf{w}_c^1, \mathbf{w}_c^2 \ldots, \mathbf{w}_c^j, \ldots, \mathbf{w}_c^d)$ de dimension d. Les référents de la carte sont représentés par $\mathcal{W} = \{\mathbf{w}_c, \mathbf{w}_c \in \Re^{\mathcal{S}}\}_{c=1}^{K}$. Chaque référent est associé à un sous-ensemble de données affectées à la cellule c, qui est noté P_c. L'ensemble des sous-ensembles forme la partition de l'ensemble des données $\mathcal{D}$, $\mathcal{P} = \{P_1, \ldots, P_c, \ldots, P_C\}$. La fonction de coût à minimiser est donc :

$$\mathcal{J}_{SOM}(\mathcal{W}, \phi) = \sum_{i=1}^{n} \sum_{c=1}^{K} K^T(\delta(\phi(\mathbf{x_i}), c)) \|\mathbf{w_c} - \mathbf{x_i}\|^2) \tag{5.1}$$

La notion de voisinage est introduite par la fonction $\mathcal{K}^T(\delta) = e^{\frac{-\delta}{T}}$.
ϕ affecte chaque observation $\mathbf{x}_i$ à une cellule unique de la carte.

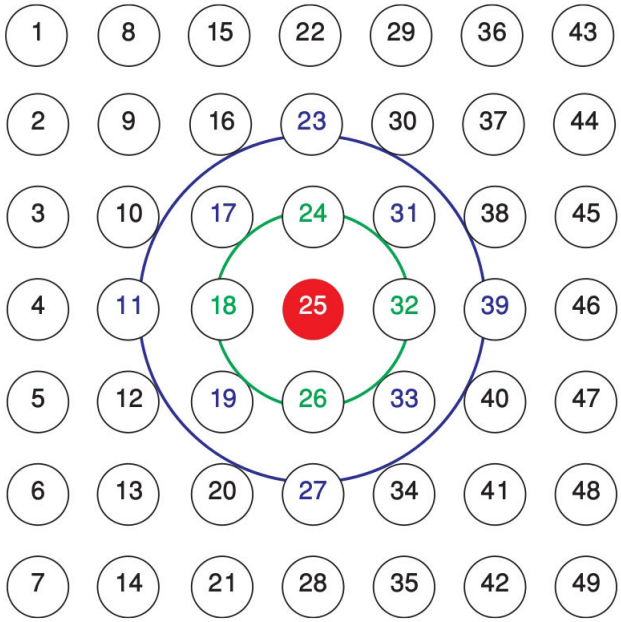


FIGURE 5.1 – Grille carrée d'une carte SOM de 49 cellules

Suivant l'algorithme des cartes SOM stochastiques, la minimisation de la fonction $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ (voir formule 5.1), pour une valeur de T fixée, est réalisée par des itérations successives, chacune se décomposant en deux phases. La première phase affecte l'ensemble des observations aux référents, la seconde minimise la valeur de la fonction de coût associée à la partition.

– **Phase d'affectation**
Il s'agit, dans cette phase, de minimiser la fonction $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ par rapport à la fonction d'affectation ϕ ; à cette étape, l'ensemble des référents $\mathcal{W}$ est fixé et est égal à celui qui est calculé durant la phase précédente. La minimisation s'obtient en

affectant chaque observation $\mathbf{x}_i$ au référent $\mathbf{w}_k$ à l'aide de la fonction d'affectation ϕ :

$$\phi(\mathbf{x}_i) = \arg \min_{1 \leq c \leq K} \parallel \mathbf{x}_i - \mathbf{w}_c \parallel^2 \tag{5.2}$$

Cette phase permet de définir une partition de l'ensemble des données. Chaque observation $\mathbf{x}_i$ étant affectée au référent le plus proche au sens de la distance pondérée.

– **Phase de minimisation**
La deuxième phase de l'itération fait décroître à nouveau $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ en fonction de l'ensemble des référents $\mathcal{W}$. Il s'agit maintenant de minimiser $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ en fonction de l'ensemble des référents $\mathcal{W}$ en supposant ϕ fixée (partition fixée). La fonction $\mathcal{J}_{SOM}(\mathcal{W}, \phi)$ étant convexe par rapport aux paramètres $\mathcal{W}$, la minimisation est obtenue en utilisant la descente du gradient, l'expression permettant de minimiser la fonction objective est donnée comme suit :

$$\mathbf{w}_c(t) = \mathbf{w}_c(t-1) - \varepsilon(t) K^T(\delta(\phi(\mathbf{x}_i), c))) \left(\mathbf{w}_c(t-1) - \mathbf{x}_i\right) \tag{5.3}$$

Dans la formule 5.3, le pas du gradient $\epsilon(t)$ décroît au cours des itérations, ainsi que le paramètre T. Ceci revient à décroître le voisinage d'influence des référents.

Recherche des GRNs représentants : set_SOM

Afin de sélectionner un ensemble de N_s GRNs représentants parmi l'ensemble des candidats E, pour chaque gène de $\mathcal{G}$, nous appliquons l'algorithme stochastique de la carte SOM (voir Section 5.3.1). Cet algorithme nécessite la matrice de dissimilarité entre les GRNs candidats de l'ensemble E, notée M_disim. Pour ce faire, nous utilisons l'*Indice de Jaccard* (IJ) comme métrique de similarité. Notons que la dissimilarité peut être calculée à partir de la similarité (*i.e.*, dissimilarité = 1− similarité). Ensuite, en fonction des valeurs de M_disim, les GRNs candidats sont groupés en N_s ($N_s \leq k$) cellules formant la grille $\mathcal{C} = \{c_1, \ldots, c_{N_s}\}$. A chaque cellule c de la grille $\mathcal{C}$ est associé un vecteur référent $\mathbf{w}_c = (\mathbf{w}_c^1, \mathbf{w}_c^2 \ldots, \mathbf{w}_c^j, \ldots, \mathbf{w}_c^{|\mathcal{S}|})$ de dimension $|\mathcal{S}|$. Les référents de la carte sont représentés par $\mathcal{W} = \{\mathbf{w}_c, \mathbf{w}_c \in \Re^{\mathcal{S}}\}_{c=1}^{N_s}$. Dans notre cas, la fonction de minimisation de coût est la suivante :

$$\mathcal{J}_{SOM_{GRN}}(\mathcal{W}, \phi) = \sum_{i=1}^{|E|} \sum_{c=1}^{N_s} K^T(\delta(\phi(\mathbf{GRN_i}), c)) \|\mathbf{w_c} - \mathbf{GRN_i}\|^2) \tag{5.4}$$

Où $\mathcal{K}^T(\delta)$ est la fonction de voisinage entre deux cellules, δ est la distance entre deux cellules et ϕ affecte chaque GRN_i candidat à une cellule unique de la carte.

Une fois les référents estimés par la fonction de coût (voir équation 5.4), nous procédons à l'identification des représentants. Pour chaque cellule c, le GRN le plus proche du référent est considéré comme le représentant de la cellule, noté GRN_rep_c. Afin d'identifier ce dernier, la *distance Euclidienne* (DIS_{euc}) entre le référent et chaque GRN de la cellule est calculée, puis, le GRN_rep_c recherché est celui qui minimise cette distance :

$$GRN\_rep_c = min(DIS_{euc}(\mathbf{w}_c, GRN_c)) \tag{5.5}$$

L'algorithme 7 résume le processus d'identification des GRNs représentants.

Algorithme 7 set_SOM : l'algorithme de recherche des GRNs représentants

pour chaque gène de $\mathcal{G}$,

Entrées : Un ensemble E de GRNs candidats, le nombre de clusters k

Sorties : Un ensemble EG de N_s GRNs représentants $\subseteq E$

1 : $EG := \emptyset$

2 : $M\_disim := 1 - IJ(GRN_i, GRN_j); i, j \in \{1, \ldots, |E|\}$

3 : $\{\mathbf{w}_c, \mathbf{w}_c \in \Re^{\mathcal{S}}\}_{c=1}^{N_s} := \mathcal{J}_{SOM_{GRN}}(\mathcal{W}, \phi)$

4 : $EG = \sqcup_{i=1}^{N_s} GRN_{rep_{c_i}}$

5.3.2 Régression linéaire

La deuxième approche de sélection numérique que nous proposons est basée sur la régression linéaire RL. Nous avons déjà utilisé cette technique dans la phase de classement des GRNs et des interactions de SelectNet et de SetNet (voir Section 4.2.3). En effet, nous avons formulé le problème de la prédiction de l'état numérique d'un gène cible g à partir des états numériques de ses régulateurs, comme étant un problème de RL. Les coefficients des régulateurs appris par la RL ont servi pour le tri des relations entre un gène cible et chacun de ses régulateurs.

Rappelons l'équation 4.7 du passage d'un GRN à une RL, où chaque gène g est approximé par $\hat{g}$, qui est une fonction linéaire de m régulateurs $r_{i(i=1,\ldots,m)}$:

$$\hat{g} = \alpha_1 r_1 + \alpha_2 r_2 + ... + \alpha_m r_m + \mu$$

Nous proposons d'attribuer un score numérique à chaque GRNs en fonction de l'erreur de prédiction du modèle de régression RL. Le score des GRNs candidats (*i.e.*, l'ensemble E) pour la régulation d'un gène donné peut être établi sur la base de la mesure de l'*erreur quadratique moyenne* (RMSE, pour Root Mean Square Error) entre le profil d'expression réel du gène g et celui estimé par ses régulateurs candidats selon le modèle de RL, $\hat{g}$:

$$RMSE(GRN) = \sqrt{\sum_{i=1}^{|\mathcal{S}|} \frac{(\hat{g}_i - g_i)^2}{|\mathcal{S}|}} \tag{5.6}$$

où $|\mathcal{S}|$ est le nombre d'échantillons d'expression de gènes ($\mathcal{S} = s_1, \ldots, s_n$).

Une fois le score RMSE calculé pour chaque GRN candidat à la régulation d'un gène, nous procédons à la sélection du/des meilleurs GRNs. Notons que deux stratégies sont possibles :

Sélection du meilleur GRN Cette stratégie consiste à sélectionner le GRN qui a la plus faible valeur de RMSE. Cependant, nous devons impérativement vérifier que cette valeur est discriminante : un seul ou-au pire-un ensemble très réduit de GRNs, possède une valeur minimale.

Sélection ensembliste Dans ce cas, nous sélectionnons un ensemble de GRNs suivant la fonction de score mise en place dans SetNet (voir l'équation 4.1 de la section 4.2.2). Cette fonction cherche l'ensemble de GRNs le plus précis parmi les GRNs candidats (*i.e.*, $\theta = 1$, la diversité entre les GRNs est assurée par le bagging). Contrairement à SetNet où la précision est mesurée sur la base du programme discret RP, dans ce cadre numérique, la précision de l'ensemble est mesurée par l'erreur RMSE. Étant donné un ensemble de GRNs, chaque gène g est approximé par $\hat{g}$, qui est une fonction linéaire des régulateurs r appartenant aux différents GRNs de l'ensemble. Par exemple, pour un ensemble de k GRNs de m_k régulateurs r_k, la précision est calculée en estimant l'erreur RMSE entre l'état résultant $\hat{g}$ et l'état du gène cible g. L'état résultant est obtenu par la formule suivante :

$$\hat{g} = (\alpha_{11} r_{11} + \cdots + \alpha_{m1} r_{m1}) + \cdots + (\alpha_{1k} r_{1k} + \cdots + \alpha_{mk} r_{mk}) + \mu \tag{5.7}$$

où μ représente le terme d'erreur (perturbation).

La capacité discriminante du score RMSE

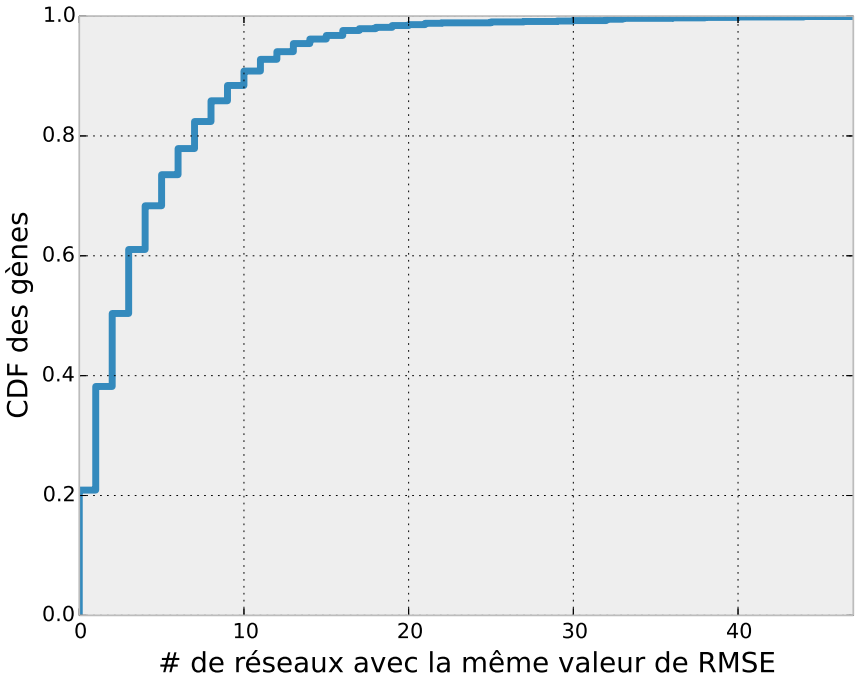


Figure 5.2 – CDF du nombre de gènes ayant des GRNs candidats de même RMSE

En vue de pouvoir implémenter la première approche de sélection (*i.e.*, sélection du meilleur GRN), nous devons quantifier la capacité discriminante de la RMSE. En effet, supposons que n GRNs possèdent la même erreur pour un gène g, le choix du meilleur peut être exprimé comme un tirage *aléatoire* dans cet ensemble de n GRNs *i.e.*, la probabilité de

choisir le meilleur est égale à $\frac{1}{n}$. Ainsi, il faut impérativement que peu de GRNs possèdent la même valeur d'erreur.

Afin de quantifier cette valeur, nous traçons la fonction de répartition de l'erreur RMSE dans la Figure 5.2 : l'axe des abscisses représente le nombre de GRNs, parmi 100 GRNs candidats, ayant une même valeur de RMSE pour un gène g alors que l'axe des ordonnées représente le pourcentage des gènes (en cumulatif). Tout d'abord, nous remarquons que pour 21% des gènes, la valeur de l'erreur est unique *i.e.*, un seul GRN possède une valeur minimale. Ensuite, pour 60% des gènes cette valeur minimale concerne au plus trois GRNs. Finalement, plus de 90% des gènes ont au plus 10 GRNs ayant la même erreur. Ces résultats indiquent que la valeur de la RMSE est discriminante et peut être utilisée pour sélectionner le GRN le plus performant.

5.4 Choix de la méthode de sélection

Dans ce chapitre, nous avons proposé deux méthodes numériques de sélection de GRNs pour H_SetNet (voir Section 5.3), néanmoins, nous devons choisir la plus performante. Pour ce faire, nous devons non seulement évaluer la performance de chacune d'elles mais aussi des méthodes de sélection initiales (*i.e.*, celle de Licorn et de SetNet). Ainsi, sans avoir recours au bagging, nous réalisons une étude comparative sur l'ensemble des données Dream5 (voir la section 3.4) pour les différentes approches de sélection :

- *best_MAE* : l'utilisation de MAE comme score discret local et la sélection d'un seul GRN(*i.e.*, la version initiale de Licorn)
- *set_MAE* : l'utilisation de la fonction de score discrète de sélection d'un ensemble de GRNs de SetNet ($\theta = 1$).
- *set_SOM* : la sélection d'un ensemble de GRNs correspondant aux représentants des clusters. Le nombre Ns de cellules est fixé à 10.
- *best_RMSE* : la sélection du meilleur GRN ayant la valeur d'erreur RMSE minimale.
- *set_RMSE* : l'utilisation de la fonction de score pour la sélection d'un ensemble de GRNs ayant la valeur d'erreur RMSE minimale.

Les résultats sont illustrés par la Figure 5.3 et le tableau 5.1.

Méthodes de sélection	Dream5	
	AUROC	AUPR
best_MAE	0.50	0.01
set_MAE	0.55	0.08
set_SOM	0.52	0.04
best_RMSE	**0.57**	**0.13**
set_RMSE	**0.59**	**0.15**

Table 5.1 – Les mesures AUROC et AUPR des différents processus de sélection discrets et numériques sur Dream5

Tout d'abord, nous constatons que l'approche numérique n'est pas *toujours* meilleure que l'approche discrète. En effet, les courbes ROC et PR de la Figure 5.3 ainsi que les faibles valeurs de AUROC et AUPR du tableau 5.1 montrent que la sélection discrète selon

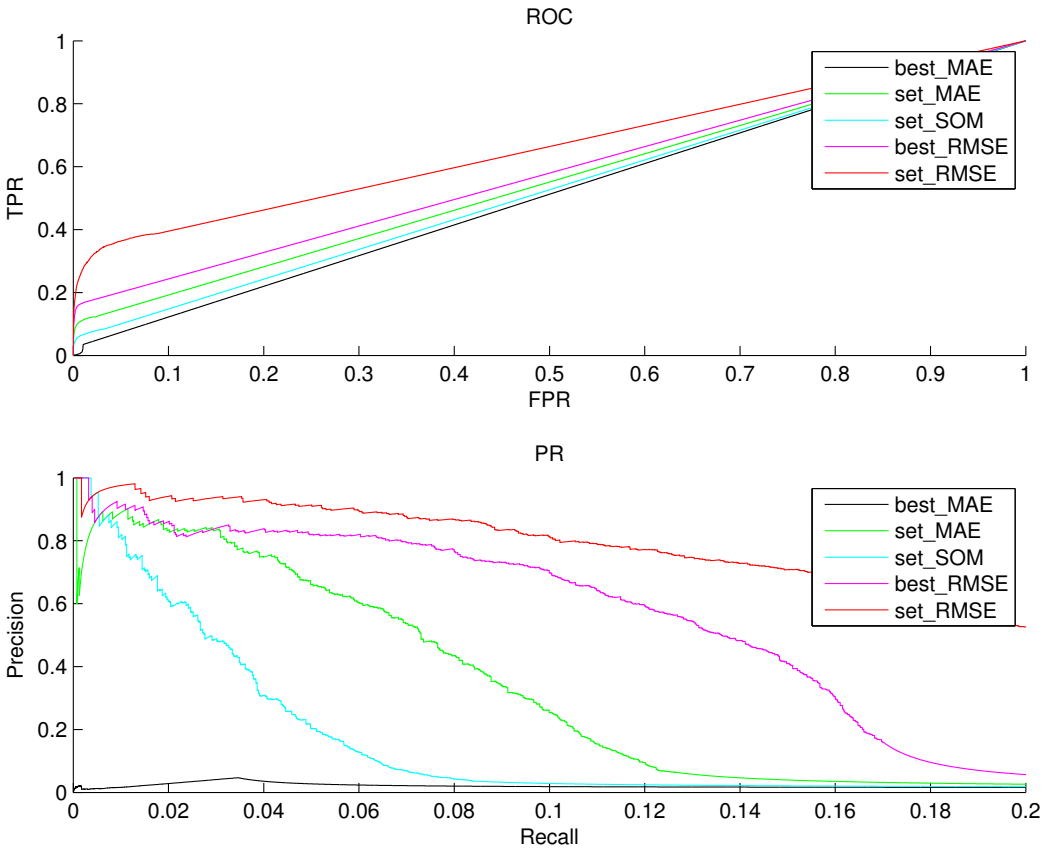


FIGURE 5.3 – Les courbes ROC et PR des différents processus de sélection discrets et numériques sur DREAM5

la *set_MAE* (courbe verte) est meilleure que celle obtenue par une approche numérique *i.e.*, *set_SOM* (courbe turquoise). Une première explication est que *set_SOM* regroupe les GRNs uniquement en trois clusters (en moyenne) alors que la *set_MAE* en sélectionne 6 en moyenne. Ainsi, ce faible nombre de GRNs sélectionnés influe négativement sur le nombre de TP trouvées. La deuxième explication repose sur l'exploitation excusive du critère de structure des GRNs pour la recherche des GRNs représentants (*i.e.*, non supervisé).

Ensuite, nous remarquons que les meilleures performances sont obtenues en utilisant les sélections *best_RMSE* et *set_RMSE* (respectivement les courbes magenta et rouge de la Figure 5.3). Il est important de noter trois points :

- la sélection d'un *seul* GRN grâce à la valeur de la RMSE est nettement supérieure à celle utilisant la valeur de la MAE particulièrement si nous comparons la precision et le rappel : *best_RMSE* est 13 fois supérieure a *best_MAE* (AUPR)
- la sélection de *best_RMSE* d'un *seul* GRN permet de réduire sensiblement les faux positifs, d'où une meilleure performance en terme d'AUPR face aux sélections ensemblistes de *set_score* et de *set_SOM*.
- la sélection ensembliste *set_RMSE* a la meilleure performance, elle est supérieure à la sélection *best_RMSE*. Ceci peut être expliqué par le fait qu'une sélection d'ensemble de GRNs permet de découvrir de nouvelles interactions assurant de meilleures performances par rapport à un seul GRN. De plus, les interactions

prédites par *best_RMSE* sont nécessairement incluses dans *set_RMSE* (*i.e.*,, la première sélectionne le meilleur GRN et la deuxième sélectionne non seulement ce meilleur GRN mais aussi d'autres GRNs qui minimisent ensemble l'erreur RMSE). Néanmoins, combiner la sélection d'ensemble avec le bagging (*i.e.*, voir l'algorithme 6 de H_SetNet) peut augmenter le nombre de TP mais aussi le nombre de FP que nous cherchons à réduire.

A ce stade, nous ne pouvons pas encore choisir entre *best_RMSE* et *set_RMSE* comme méthode de sélection. Une analyse plus approfondie des deux approches doit être entreprise.

5.5 H_SetNet^1 Vs H_SetNet^2

Tout d'abord, rappelons que H_SetNet utilise le bagging — une méthode d'ensemble — afin d'introduire de la diversité. Ainsi, pour pouvoir choisir entre les deux algorithmes de sélection, nous devons comparer les deux versions possible de H_SetNet :

- H_SetNet^1 : Cette version utilise la sélection *best_RMSE*. Ainsi, cet algorithme utilise une seule méthode d'ensemble : le bagging.
- H_SetNet^2 : Cette version utilise la sélection *set_RMSE*. Ainsi, cet algorithme utilise deux méthodes d'ensemble : (i) la sélection d'un ensemble de GRNs de *set_RMSE* et (ii) le bagging

Les résultats de cette comparaison sont illustrés par la Figure 5.4 et les tableaux 5.2 et 5.3.

Ces résultats montrent des performances similaires pour les deux méthodes. Les courbes ROC et PR superposées indiquent un comportement quasi-identique. Ceci est confirmé par les mesures AUROC et AUPR du tableau 5.2 montrant une faible différence de l'ordre de 10^{-3}. Cependant, la version H_SetNet^1 est moins couteuse en temps de calcul que H_SetNet^2 (voir tableau 5.3).

Method	Dream5	
	AUROC	AUPR
H_SetNet^1	0.70141600	0.23555911
H_SetNet^2	0.70428973	0.23874097

Table 5.2 – Les mesures AUROC et AUPR de H_SetNet^2 et H_SetNet^1

Method	Dream5
	temps de calcul (mn.s)
H_SetNet^1	$3.43_{(0.07)}$
H_SetNet^2	$5.38_{(0.23)}$

Table 5.3 – Le temps moyen de calcul par gène de H_SetNet^1 et H_SetNet^2 sur les données Dream5 (Le calcul est estimé par une machine 32 bits Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz).

Par ailleurs, en analysant plus précisément les GRN inférés, H_SetNet^2 sélectionne en moyenne deux GRNs par gène cible mais infère des réseaux de mêmes performances

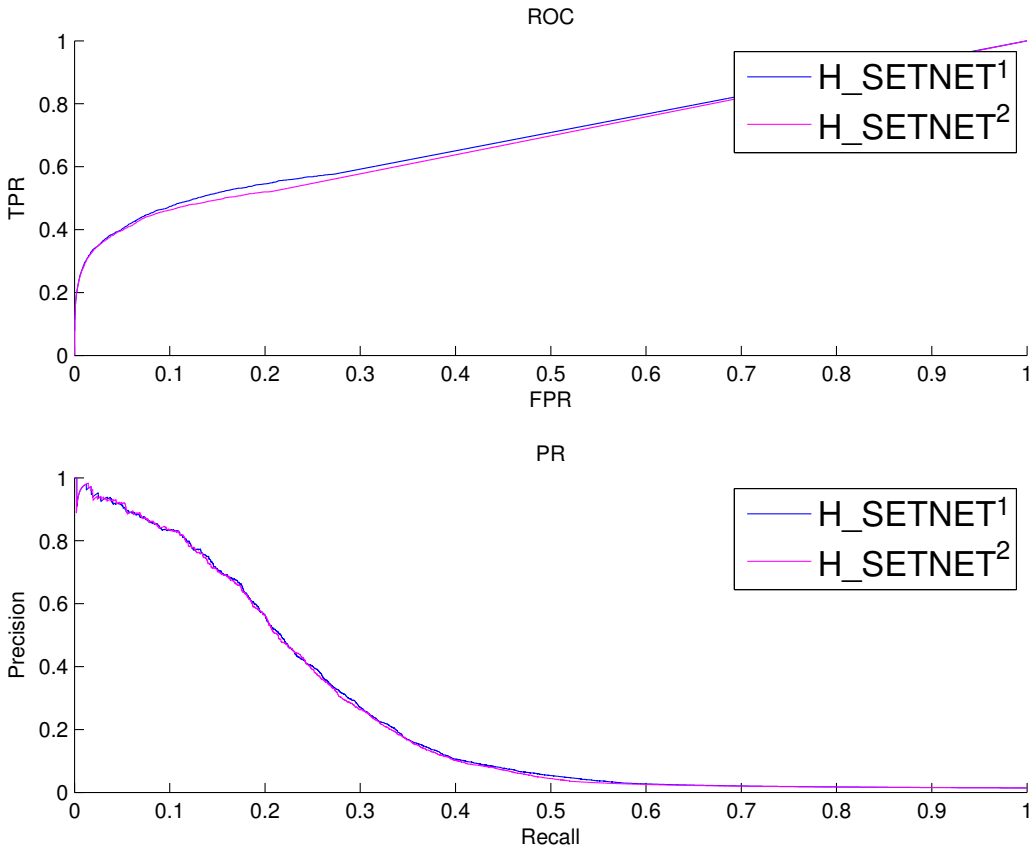


FIGURE 5.4 – Les courbes ROC et PR de H_SETNET[1] et H_SETNET[2] SUR DREAM5

que H_SETNET[1] qui ne sélectionne qu'un seul GRN. Pour raffiner cette constatation, nous quantifions le nombres d'interactions TP et FP présentes dans les GRNs inférés par ces deux méthodes en calculant leurs intersections avec les interactions réelles de la base *Gold*. Les résultats sont présentés par le tableau 5.4. Nous constatons que pour un gain d'environ 50 TP, H_SETNET[2] introduit 3000 FP de plus que H_SETNET[1]. Cela peut être expliqué par le fait que l'étape de bagging introduit assez de diversité et donc de vrais positifs rendant la seconde étape — celle de sélection d'ensemble de GRNs d'un même boostrap — inutile, voire inadéquate car elle n'introduit quasiment que de faux positifs.

Method	DREAM5	
	TP	FP
H_SETNET[1]	2309	89050
H_SETNET[2]	2359	92050

TABLE 5.4 – Le nombre d'interactions prédites (TP et FP) par H_SETNET[1] et H_SETNET[2] sur les données DREAM5

Pour conclure, nous adoptons la version H_SETNET[1] pour deux raisons :

– l'introduction d'un grand nombre de faux positifs par H_SetNet^2 pour un faible gain en vrais positifs en comparaison à H_SetNet^1.

– la complexité de la sélection gloutonne de H_SetNet[2] qui demande un temps de calcul largement supérieur à celui de H_SetNet[1] pour un faible gain en performance.

H_SetNet est par conséquent définie par la sélection du meilleur GRN ayant l'erreur RMSE minimal.

5.6 Expérimentations et résultats

Nous procédons dans cette section à l'expérimentation de la méthode hybride H_SetNet. Suivant la même méthodologie que dans le chapitre précédent (voir la section 3.4), nous évaluons, dans un premier lieu, l'amélioration de performance de H_SetNet par rapport à SetNet (*i.e.*, gain en TP, baisse de FP). Ensuite, nous effectuons une analyse comparative de H_SetNet avec deux algorithmes de l'état d'art : Aracne et Genie3.

5.6.1 Réduction des faux positifs

Rappelons que le but premier de H_SetNet est de réduire sensiblement le nombre de faux positifs en comparaison à SetNet. Ainsi, la première étude que nous menons est une analyse comparative entre H_SetNet et SetNet en terme des liens prédits. Concrètement, pour deux seuils de rappel fixés à 0.02 et 0.1, le nombre d'interactions inférées par les deux méthodes est présenté par le diagramme de Venn de la Figure 5.5.

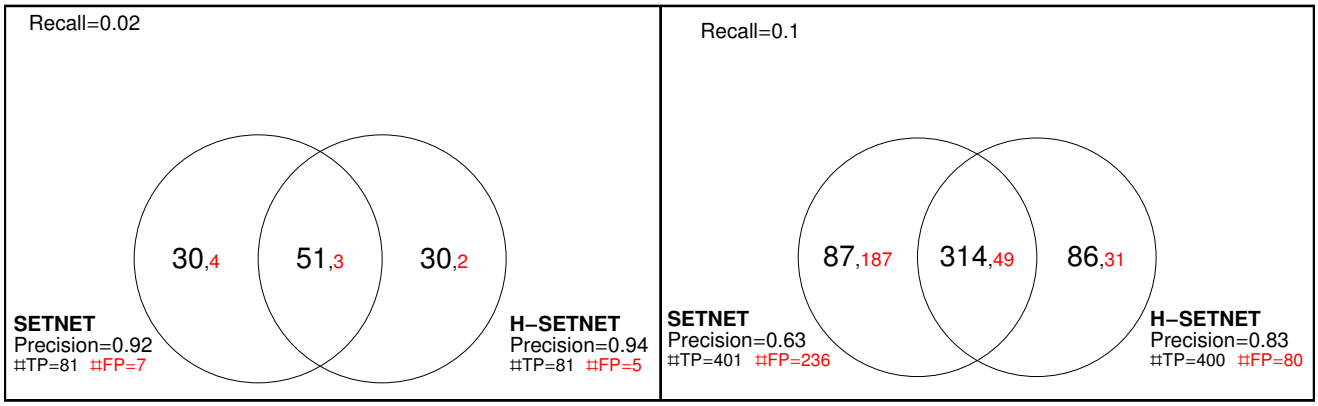


Figure 5.5 – Le nombre d'interactions correctes TP et fausses FP, respectivement en noir et en rouge, identifiées pour un rappel fixé (0.02 et 0.10) par SetNet et H_SetNet sur la base Dream5

Une première constatation est que le nombre de vrais positifs prédits par les deux méthodes, pour un rappel de 0.02, est le même (81 TP). Ceci démontre bien que H_SetNet— tout comme SetNet– sont capables d'inférer un nombre important de vrais positifs. Plus important encore, pour un rappel de 0.1, le comportement de H_SetNet est largement meilleur que celui de SetNet. En effet, le nombre de faux positifs passe de 238 à seulement 80 soit une division par trois, tout en gardant un taux de vrais positifs relativement élevé (environ 400) pour les deux méthodes. Ainsi, la sélection numérique utilisée permet de discriminer les bonnes interactions des mauvaises en minimisant le nombre de faux positifs. Par conséquent, pour les deux

seuils de précision, H_SetNet surpasse sensiblement SetNet(0.94 vs 0.92 et 0.83 vs 0.63).

Pour une analyse plus générale, nous illustrons les mesures AUROC et AUPR des deux méthodes dans le tableau 5.5. En effet, les résultats confirment que la sélection numérique améliore les performances. En sélectionnant moins de GRNs, H_SetNet favorise la détection des plus précis. Les mesures AUROC et AUPR de SetNet passent respectivement de 0.67 à 0.70 et de 0.18 à 0.23.

Method	Dream5	
	AUROC	AUPR
H_SetNet	0.70	0.23
SetNet	0.67	0.18

Table 5.5 – Les mesures AUROC et AUPR de SetNet et H_SetNet sur Dream5

La deuxième phase de cette analyse consiste à évaluer le processus de classement des interactions prédites. En effet, nous cherchons à voir comment les interactions sont triées et si les vrais positifs sont mieux classés que les faux négatifs. Pour ce faire, nous quantifions le nombre d'interactions TP pour H_SetNet et SetNet selon leurs rangs dans les liens prédits. Les résultats sont présentés par la Figure 5.6. La croissance plus rapide de la courbe de H_SetNet démontre bien que ce dernier classifie les vrais positifs avant les faux négatifs (en comparaison à SetNet). Plus précisément, pour les 1000 premières interactions TP prédites, H_SetNet les classe dans les 2500 premiers rangs alors que SetNet les classe dans les 6000 premiers générant ainsi 3500 faux positifs. Ces deux expérimentions ont permis de mettre en évidence la capacité de H_SetNet à discriminer les interactions TP des FP et ainsi surpasser SetNet.

5.6.2 Comparatif de performances

Dans ce qui suit, nous comparons H_SetNet aux deux algorithmes : Aracne et Genie3. Premièrement, pour l'analyse des liens prédits, nous suivons le même protocole expérimental que précédemment. Les résultats sont présentés par le diagramme de Venn de la Figure 5.7.

Nous observons que, pour les deux seuils de rappel, le nombre de TP inférés est quasiment le même pour les trois méthodes (environ 80 pour un rappel de 0.02 et 400 pour un rappel de 0.1). En plus, pour un rappel de 0.1, H_SetNet a l'avantage d'inférer le minimum de FP (uniquement 80 contre 92 pour Genie3 et 253 pour Aracne). Un avantage plus intéressant, H_SetNet est la méthode qui infère le plus d'interactions réelles non identifiées par aucune des autres méthodes (environ 168).

Par ailleurs, pour évaluer la performance des différentes méthodes, nous illustrons les résultats dans le tableau 5.6 et la Figure 5.8.

Ces résultats montrent que H_SetNet présente des performances comparables à celles de Genie3 et meilleures que celles d'Aracne. En effet, d'après la Figure 5.8, H_SetNet infère légèrement moins d'interactions TP que Genie3 (voir courbes ROC).

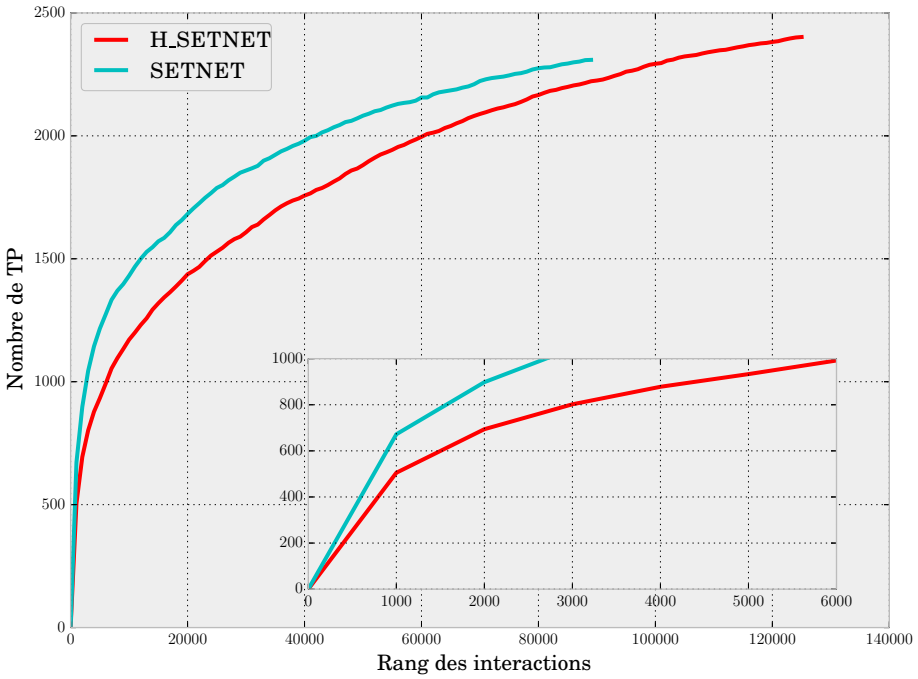


FIGURE 5.6 – Le nombre de TP inférés par SETNET et H_SETNET suivant le classement d'importance des interactions

Method	DREAM5	
	AUROC	AUPR
H_SETNET	0.70	0.23
ARACNE	0.55	0.10
GENIE3	**0.73**	**0.26**

TABLE 5.6 – Les mesures AUROC et AUPR des différentes méthodes appliquées sur DREAM5

Cependant, il arrive à les mieux classer et à assurer une meilleure précision pour un rappel entre 0 et 0.15 (voir courbes PR de la Figure 5.8).

Ces résultats mettent en évidence la supériorité des méthodes d'ensemble dans le contexte d'inférence des GRNs. En effet, les deux méthodes ayant les meilleures performances sont GENIE3 et notre méthode hybride H_SETNET (voir tableau 5.6).

Bien que ces résultats soient très encourageants, une contrainte supplémentaire freine le passage à l'échelle de ces algorithmes à une application sur des données humaines. En effet, toutes les expérimentations sont menées sur la base DREAM5 ou le nombre d'échantillons est relativement important (*i.e.*, un jeux de données de 805 échantillons pour 1643 gènes (voir Tableau 3.1). Or, dans le scénario humain, le nombre d'échantillons peut être petit (*i.e.*, jusqu'à *deux ordres de grandeur* plus petit que celui des gènes). Dans ce qui suit, nous analysons l'impact de la *contrainte de sous échantillonnage* sur les performances de chacun de ces algorithmes. Cette étape est primordiale avant le passage aux données

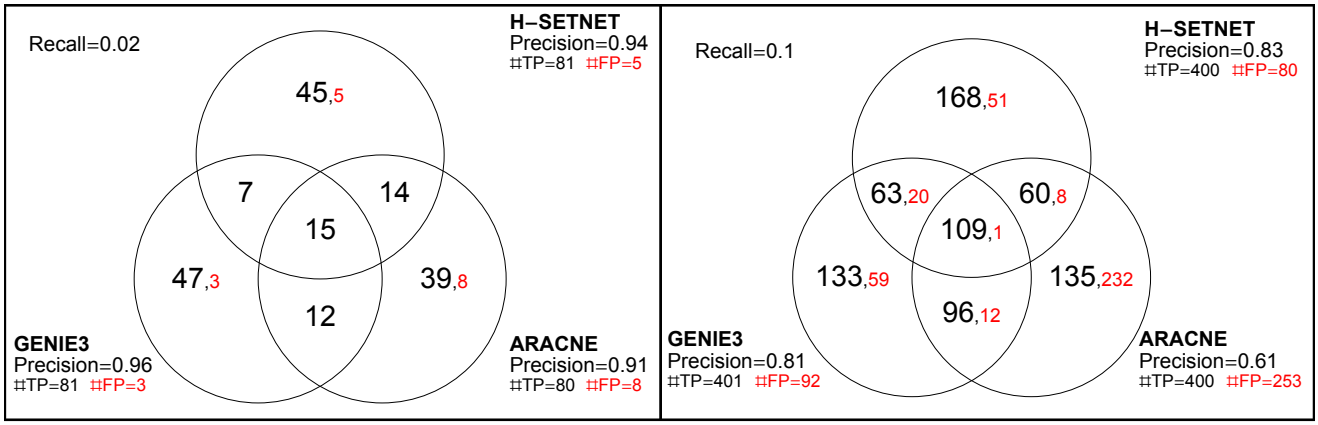


FIGURE 5.7 – Le nombre des interactions correctes TP et fausses FP, respectivement en noir et en rouge, identifiées pour un rappel fixé (0.02 et 0.10) par SETNET, ARACNE et GENIE3 sur la base DREAM5

humaines.

5.7 Robustesse au sous échantillonnage

Cette section analyse la robustesse de H_SETNET, ARACNE et GENIE3 face à la contrainte de sous échantillonnage. Tout d'abord, définissons *la robustesse* d'une méthode d'inférence. *Par une méthode* ***robuste****, nous entendons qu'elle est capable de préserver des performances similaires face à la contrainte du sous échantillonnage.*

La sensibilité caractérisant les méthodes d'inférence sur les données de grande dimension [BS09] est un aspect important qui a commencé à attirer l'attention de la communauté scientifique. En effet, lorsque nous traitons des données biologiques, la taille des échantillons disponibles est souvent extrêmement faible par rapport au nombre de gènes [Dra03, YSL07], ce qui oblige les algorithmes d'inférence à prendre en considération cette contrainte supplémentaire.

Pour simuler cette situation, nous sélectionnons des sous-échantillons aléatoires de petite taille n, $n \in \{50, 200, 400\}$) tel que $n << |\mathcal{S}|$ de l'ensemble de données DREAM5. Ainsi, d'une taille initiale de 805 échantillons, nous échantillonnons des sous-ensembles de données notés DREAM5sub$n$ ($n \in \{50, 200, 400\}$)). La phase de test consiste à appliquer chaque algorithme à un échantillon de taille n pour vérifier la sensibilité de ces performances. Cette étape est répétée 10 fois afin de garantir la fiabilité des résultats. Les tableaux 5.7, 5.8 et 5.9 fournissent la moyenne et l'écart-type de ces mesures.

Nous choisissons de présenter les courbes ROC et PR des différentes méthodes sur le sous-échantillon le plus critique, DREAM5sub50. Comme pour tous les sous-échantillons, nous avons réalisé 10 fois l'expérimentation et nous avons mesuré la moyenne des 10 résulats obtenus. Le sous-ensemble ayant les performances des méthodes les plus proches de la moyenne des 10 expérimentations est illustré dans la Figure 5.9.

Comme prévu, les tableaux 5.7, 5.8 et 5.9 montrent que lorsque la taille de l'échantillon

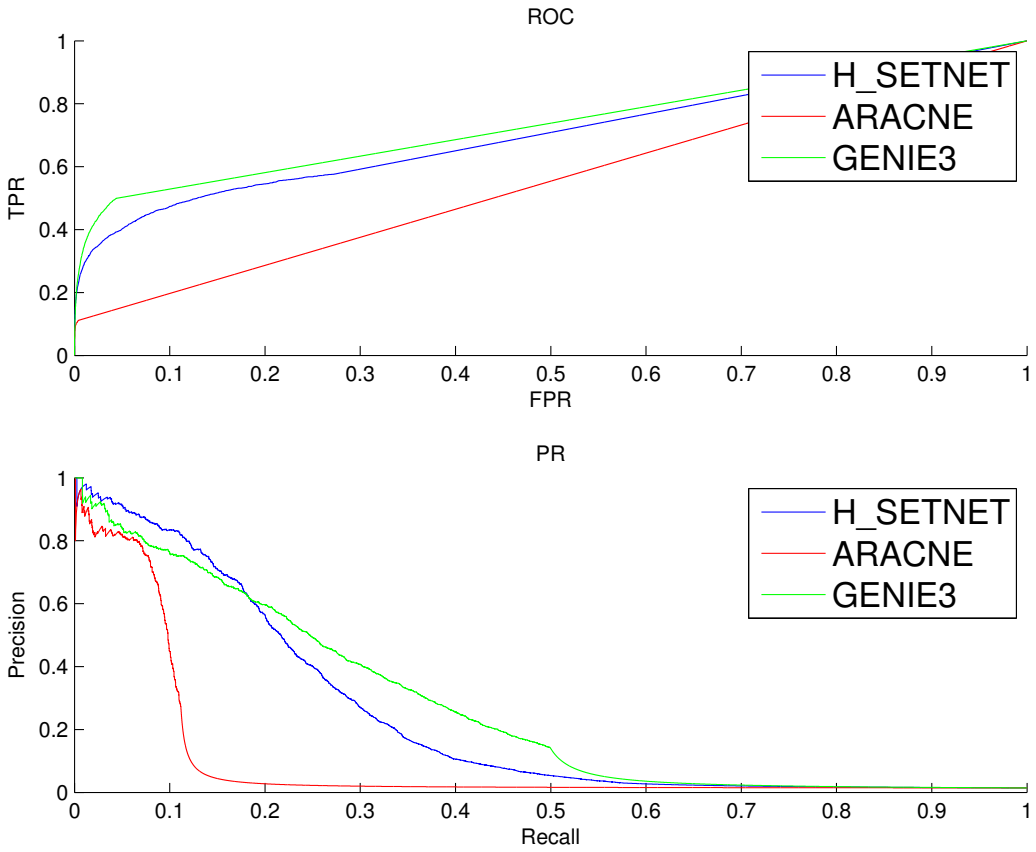


FIGURE 5.8 – Les courbes ROC et PR des différentes méthodes appliquées sur DREAM5

Method	10 DREAM5sub400	
	AUROC $\mu(\sigma)$	AUPR $\mu(\sigma)$
H_SETNET	**0.69**$_{(0.006)}$	0.21$_{(0.007)}$
ARACNE	0.55$_{(0.001)}$	0.10$_{(0.003)}$
GENIE3	0.68$_{(0.002)}$	**0.23**$_{(0.004)}$

TABLE 5.7 – La moyenne μ et l'écart type (σ) des mesures AUROC et AUPR des 10 sous-ensembles DREAM5sub400

Method	10 DREAM5sub200	
	AUROC $\mu(\sigma)$	AUPR $\mu(\sigma)$
H_SETNET	**0.67**$_{(0.006)}$	**0.21**$_{(0.013)}$
ARACNE	0.54$_{(0.002)}$	0.08$_{(0.005)}$
GENIE3	0.66$_{(0.005)}$	0.20$_{(0.01)}$

TABLE 5.8 – La moyenne μ et l'écart type (σ) des mesures AUROC et AUPR des 10 sous-ensembles DREAM5sub200

diminue, l'écart type de toutes les mesures de performance augmente et les moyennes des mesures AUROC et AUPR diminuent pour toutes les méthodes. Cependant, ces

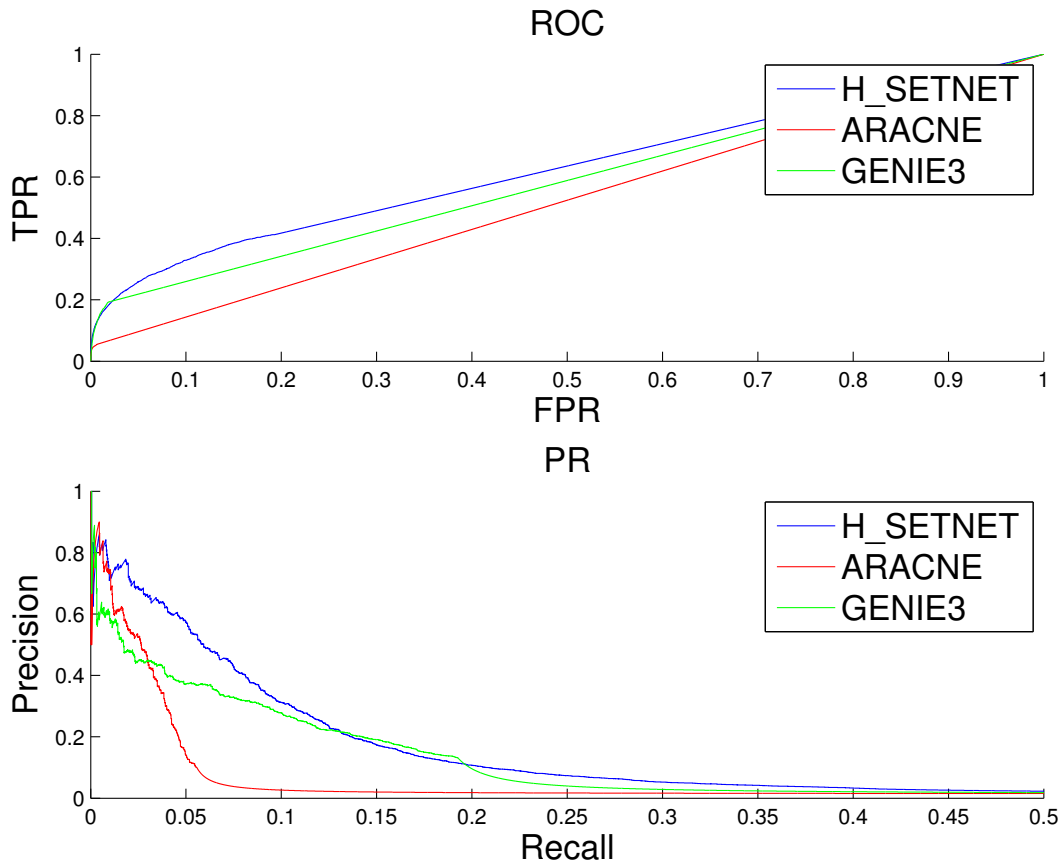


FIGURE 5.9 – Les courbes ROC et PR du sous-ensemble le plus proche à la moyenne des 10 DREAM5sub50

Method	10 Dream5sub50	
	AUROC $\mu(\sigma)$	AUPR $\mu(\sigma)$
H_SetNet	$\mathbf{0.63}_{(0.02)}$	$\mathbf{0.10}_{(0.012)}$
Aracne	$0.52_{(0.003)}$	$0.04_{(0.005)}$
Genie3	$0.58_{(0.01)}$	$0.08_{(0.016)}$

Table 5.9 – La moyenne μ et l'écart type (σ) des mesures AUROC et AUPR des 10 sous-ensembles Dream5sub50

dégradations de performances varient d'une méthode à une autre.

Ces résultats indiquent que H_SetNet surpasse Aracne sur les petits sous-échantillons. En plus, il est comparable à Genie3 sur Dream5sub400 et bien meilleur sur Dream5sub200 et Dream5sub50. De 805 échantillons à 50, les performances de H_SetNet se dégradent lentement. Précisément, la valeur de l'AUROC passe de $0,70$ à $0,63$ et celle de l'AUPR de $0,23$ à $0,10$. Par contre, le rendement de Genie3 se dégrade rapidement, il passe de $0,73$ à $0,59$ pour le AUROC et de $0,26$ à $0,08$ pour le AUPR. Par ailleurs, nous constatons une légère diminution pour Aracne.

Sous la forte contrainte du sous échantillonnage, principalement dans Dream5sub200 et Dream5sub50, H_SetNet a les meilleurs résultats. En comparant les courbes ROC de la Figure 5.6 de Dream5 (805 échantillons) et les courbes ROC de la Figure 5.9 de Dream5sub50, nous observons le changement de comportement de Genie3 qui perd en performances. Nous pouvons donc confirmer que H_SetNet est *robuste* et continue à avoir de bons résultats sous la forte contrainte de dimension.

D'après la littérature [Bre01, Die00b, MMHLS09, MO11], les deux méthodes d'ensemble employées par H_SetNet et Genie3, respectivement le bagging et les forêts aléatoires, résistent aux bruits. Toutefois, en se comparant au bagging [LLPS05, DUDA06], les forêts aléatoires détiennent l'avantage de l'utilisation de plusieurs sous-ensembles de variables, adaptée pour les données de grande dimension. D'autres travaux [LLPS05, DUDA06] affirment que les forêts aléatoires sont en mesure de préserver la précision prédictive et présentent des résultats meilleurs que le bagging.

Par contre, dans notre cas d'étude, H_SetNet présente des performances de robustesse meilleures que celles des forêts aléatoires de Genie3. En effet, Genie3 est une méthode purement numérique, alors que H_SetNet est hybride. En effet, il génère les GRNs candidats potentiels à la régulation d'un gène cible sur la base d'un modèle de régulation discret RP et il choisit les GRNs en s'appuyant sur le modèle numérique de régression linéaire. Par conséquent, nous expliquons les bonnes performances de H_SetNet par le couplage du modèle discret avec le modèle numérique. La combinaison de la robustesse du discret avec la précision du numérique dans une seule méthode engendre une méthode plus robuste qu'une méthode purement numérique tel que Genie3.

5.8 Inférence de la régulation des gènes dans les cancers humains : Cancer de la vessie

Dans le domaine du cancer, la technologie des puces à ADN est considérée comme un outil de diagnostic prometteur, car de nombreux gènes sont exprimés différemment entre les échantillons tumoraux et normaux. Néanmoins, les causes exactes de cette différence ne sont pas clairement identifiées.

Beaucoup d'oncogènes agissent en tant que facteurs de transcription eux-mêmes ou ont un effet direct sur des facteurs de transcription. Quand ces oncogènes sont activés par un mécanisme génétique (mutation, amplification ou translocation), ils activent ou désactivent, directement ou indirectement, plusieurs facteurs de transcription qui changent le niveau d'expression de leurs gènes cibles par des mécanismes complexes. Cependant, le niveau d'expression de chaque gène cible est directement lié au niveau d'expression de ses régulateurs. Plusieurs études ont récemment démontré la capacité des méthodes d'inférence de réseau à identifier les gènes qui sont des conducteurs de cancer [LRA+10].

Cette section s'inscrit dans ce cadre d'études expérimentales sur des données *humaines* d'expressions de cancer de la vessie. En effet, nous expérimentons la capacité de notre méthode d'inférence H_SetNet à extraire les réseaux de régulation responsables de cette maladie.

Dans la première partie de cette section, nous présentons les données utilisées et l'étape de discrétisation. Les parties qui suivent sont consacrées à l'évaluation des relations de régulations inférées, l'analyse de la coopération des régulations inférées et l'analyse des GRNs inférés.

Notons que les expérimentations de cette section sont réalisées en collaboration avec Rémy Nicol et Mohamed Elati de l'institut de Biologie Systémique et Synthétique (iSSB).

5.8.1 Préparation des données

Dans cette étude, nous utilisons des données humaines de 85 échantillons [Sa06] : 80 échantillons de carcinomes de vessie à différents stades de la maladie et 5 échantillons normaux (urothélium normal de sujets dépourvus de cancer). Ces échantillons ont été collectés à l'hôpital Henri Mondor à Créteil. Les puces à ADN utilisées sont des puces Affymetrix de type HG-U95A.

Nous nous intéressons à trois tendances de variation du niveau d'expression d'un gène : dans un échantillon tumoral donné, le niveau d'expression d'un gène peut *augmenter* par rapport à son niveau *normal* (noté par 1 dans la matrice discrétisée) ; il peut *diminuer* par rapport à son niveau normal (noté par -1) ; ou encore être *stable* (noté par 0). Nous définissons alors la technique suivante de discrétisation afin de transformer la matrice originale O en une matrice discrète M qui code, pour chaque gène g, la tendance de variation du niveau d'expression dans les échantillons tumoraux par rapport au niveau normal, calculé comme la moyenne du niveau d'expression de g dans les échantillons normaux.

Soit $\mu_n(g)$ et $\sigma_n(g)$ la moyenne et l'écart type du niveau d'expression de g dans les échantillons normaux et ρ un paramètre fixé, entre autres, par la densité de la matrice

souhaitée :

$$M(g, s_i) \;=\; \begin{cases} 1, & O(g, s_i) > (\mu_n(g) + \rho \cdot \sigma_n(g)) ; \\ \text{-1}, & O(g, s_i) < (\mu_n(g) \text{ - } \rho \cdot \sigma_n(g)) ; \\ 0, & \text{sinon.} \end{cases} \tag{5.8}$$

Le paramètre ρ de discrétisation est fixé à 2, ce qui correspond à une densité de 30% de la matrice discrétisée. Afin de choisir une liste de facteurs de transcription, nous sélectionons parmi les gènes représentés sur les puces ADN les facteurs de transcription dans la base TRANSFAC®[2] [MFG+03]. Cette sélection est réalisée en utilisant le symbole standard du gène ou les identificateurs SwissProt, si disponibles, ce qui nous amène à extraire un ensemble de 699 facteurs de transcription et 7224 gènes. Notons que le nombre de gènes est beaucoup plus grand que le nombre d'échantillons (*i.e.*, 85).

Contrairement à l'évaluation des méthodes d'inférence sur DREAM5, il n'existe aucune base Gold disponible pour les données humaines. Cependant, dans le contexte de l'inférence des réseaux de régulation, plusieurs types de données, que nous appelons *évidences externes*, peuvent compléter et évaluer les prédictions. Ces évidences externes sont sous la forme de listes d'interactions possibles extraites de la base TRANSFAC® [MKMF+06].

Notre but est de comparer les interactions de régulation prédites avec les évidences externes pour tester si l'intersection est significativement plus grande qu'une estimation aléatoire. Pour cela, nous extrayons deux types d'évidences externes de TRANSFAC ® :

- *Sites de fixation de facteurs de transcription TFBS :* Les TFBS (en anglais Transcription Factor Binding Sites) sont de courtes séquences d'ADN sur lesquelles peuvent se fixer des facteurs de transcription. Conjointement, les TFBS et leurs facteurs de transcription contrôlent l'expression des gènes. Ils vont contrôler l'initiation de la transcription, et activer ou inhiber l'expression des gènes qu'ils régulent dans un système complexe pouvant comprendre plusieurs cascades de régulation et qui est régulé à différents niveaux hiérarchiques [VDFV03].
  Dans notre cas, les évidences externes sont le résultat de la numérisation des promoteurs de gènes avec les modèles TFBS pour 76 facteurs de transcription, ce qui représente 39% des facteurs de transcription dans l'ensemble de données. Notons que l'utilisation de ces TFBS n'est pas forcément très fiable, mais c'est une indication sur les interactions possibles (facteurs de transcription – gènes cible).
- *Immuno-précipitation de la chromatine* ChIP (en anglais Chromatin Immuno-Precipitation) est une technique mise au point en 2000 pour mettre en évidence les sites de fixation de protéines d'intérêt comme les facteurs de transcription [Orl00]. Cette technique permet d'isoler des fragments d'ADN qui ont interagit avec une protéine (voir Annexe A pour plus de détails). Les évidences externes extraites sont le résultat d'un séquençage à haut débit (ChIP-seq[3]) ou d'une analyse de puces

2. La base TRANSFAC® fournit des données expérimentalement prouvés sur les facteurs de transcription eucaryotes, leurs sites de liaison, des séquences de liaison consensus et les gènes régulés. Elle contient entres autres des données sur les facteurs de transcription eucaryotes qui agissent d'une manière cooperative.

3. le ChIP-seq (Chromatin Immunoprecipitation sequencing) utilisé pour identifier les sites de fixation d'une seule protéine associée à l'ADN. En ciblant une protéine particulière avec un anticorps spécifique.

ADN (ChIP-chip [4]) pour 26 facteurs de transcription, ce qui représente 13% des facteurs de transcription dans l'ensemble de données.

Nous considérons que les données ChIP sont plus proches de la réalité vu que le nombre des facteurs de transcriptions est plus faible que celui de TFBS. Par ailleurs, ces deux types d'évidences externes sont spécifiques au données humaines. Cependant, seules les données TFBS ne sont pas spécifiques à un contexte cellulaire bien défini, alors que ChIP est particulièrement spécifique à l'origine des tissus.

5.8.2 Performances des régulations inférées

Sur les données de cancer de la vessie, nous comparons les régulations inférées par H_SetNet à celles inférées par Genie3, ainsi que celles inférées par Aracne, l'algorithme le plus utilisé dans les applications en *biologie des systèmes du cancer*.

De plus, nous nous comparons à une méthode dite *baseline* qui consiste à inférer des interactions d'une manière aléatoire. Toutes les méthodes génèrent en sortie une liste d'interactions inférées classées par un score. Pour évaluer la capacité d'une méthode à inférer les relations TFBS et ChIP, nous quantifions le recouvrement des n meilleures interactions inférées. Ainsi, diverses valeurs de n sont testées (n varie de 50 à 50000). Par ailleurs, nous réalisons un test de significativité sur les interactions inférées. Pour celà, nous optons pour le *test de Fisher* d'une confiance de 95% [5].

Les tableaux 5.10 et 5.11 présentent les résultats du recouvrement des n meilleures interactions inférées par baseline, Genie3, Aracne et H_SetNet avec les données TFBS et les données ChIP.

Les résultats du tableau 5.10 montrent que même s'il est difficile de prédire les interactions TFBS, Genie3 est incapable de faire mieux que l'aléatoire (*i.e.*, baseline), alors que Aracne et H_SetNet présentent un recouvrement important lorsque n est grand (*i.e.*, $n = 5000$ interactions). En plus, d'après le test de Fisher, H_SetNet et Aracne sont les seules méthodes qui fournissent des interactions significativement enrichies (respectivement $p-value < 0.001$ et $p-value < 0.01$). Ainsi, le recouvrement des données TFBS avec les interactions inférées par H_SetNet sont beaucoup plus significatives que pour Aracne, et ce avec une valeur de p-value de $3.9.10^{-10}$.

Quant aux résultats des évidences externes ChIP, le tableau 5.11 affirme que H_SetNet est meilleur que toutes les autres méthodes pour toutes les valeurs de recouvrement n. En plus, le test statistique confirme que les interactions de H_SetNet sont significativement enrichies en interactions existantes de ChIP ($p < 0.001$ à partir des 100 meilleures interactions).

---

4. ChIP-chip combine la technologie des puces à ADN à la technique ChIP. Développée en 2000, elle permet de localiser à l'échelle génomique l'ensemble des séquences reconnues par un facteur de transcription d'intérêt [RRW+00].

5. Le test de Fisher est un test d'hypothèse statistique qui permet de tester l'égalité de deux variances (S_x et S_y) en faisant le rapport des deux variances et en vérifiant que ce rapport ne dépasse pas une certaine valeur théorique $F_{théorique}$ que l'on cherche dans la table de Fisher.

$$F_{observé} = \frac{S_x^2}{S_y^2} \quad \begin{cases} > F_{théorique}, & \text{variances trop différentes – non homogènes ;} \\ < F_{théorique}, & \text{variances proches – homogènes ;} \end{cases}$$

n interactions	**TFBS**			
	baseline	GENIE3	ARACNE	H_SETNET
50	6.3	2	3	**10 .**
100	12.7	8	4	**14**
500	**63.4**	57	27	49
1000	**126.8**	109	81	107
5000	**634.1**	556	587	565
10000	1268.2	1138	**1286**	1200
50000	6341	6086	6534 **	**6769 *****

TABLE 5.10 – Recouvrement entre les interactions inférées par baseline, GENIE3 ,ARACNE et SETNET, et les interactions TFBS. *test de Fisher* : $p < 0.1^{\bullet}$, $p < 0.05^{*}$, $p < 0.01^{**}$, $p < 0.001^{***}$ et pas de symbole pour $p \geq 0.1$.

n interactions	**ChIP**			
	baseline	GENIE3	ARACNE	H_SETNET
50	3.1	6 .	1	**9 ****
100	6.2	11 *	1	**17 *****
500	31	44 *	20	**64 *****
1000	61.9	80 *	49	**108 *****
5000	309.7	395 ***	343 *	**417 *****
10000	619.3	761 ***	762 ***	**819 *****
50000	3096.6	3380 ***	3482 ***	**3787 *****

TABLE 5.11 – Recouvrement entre les interactions inférées par baseline, GENIE3 ,ARACNE et SETNET, et les interactions ChIP. *test de Fisher* : $p < 0.1^{\bullet}$, $p < 0.05^{*}$, $p < 0.01^{**}$, $p < 0.001^{***}$ et pas de symbole pour $p \geq 0.1$.

5.8.3 Analyse de la coopération des régulations inférées

Un des avantages principaux de la phase de l'extraction de GRNs candidats (LICORN) est de modéliser explicitement les relations de coopération entre les régulateurs pour réguler un gène dans le processus d'inférence. Ce phénomène biologique est crucial dans le processus de régulation génétique. La possibilité d'inférer des co-régulateurs pour un gène cible au lieu des relations par paires peut potentiellement conduire à une amélioration des performances. Plus important encore, inférer des co-régulateurs permet de modéliser des mécanismes clés de régulation des systèmes étudiés, *e.g.*progression du cancer.

Deux gènes régulateurs sont définis en tant que régulateurs de coopératifs (*i.e.*, co-régulateurs), lorsque les protéines régulatrices codées par ces gènes sont susceptibles de former un complexe de protéine fonctionnant d'une manière coordonnée.

Par conséquent, afin d'évaluer si H_SETNET est en mesure de déduire les interactions coopératives pertinentes entre les régulateurs, nous comparons toutes les paires de co-activateur et co-inhibiteurs identifiées au moins une fois dans les GRNs appris pour le cancer de la vessie à la base de données des interactions protéiques STRING [FSF+12]

Les résultats du recouvrement sont résumés dans le tableau 5.12. Nous constatons

qu'une partie significative de la liste des co-régulateurs prédits par H_SetNet est connue dans la base des interactions protéiques STRING.

STRING			
Valide	Prédits	Aléatoire expected	$p-value$
765	4723	529.04	5.05^{-34}

Table 5.12 – Résultats du recouvrement entre les interactions coopératives prédites par H_SetNet, *i.e.*, paires de co-régulateurs, et la base de données d'interactions protéiques (STRING).

Nous avons également étudié la liste des co-régulateurs et nous avons constaté que la paire composée de FOXA1 et GATA3 fait partie des 10 paires les plus fréquentes des co-régulateurs identifiés. Ces gènes sont précédemment considérés, dans le Cancer du Sein [KLL+11], comme des régulateurs appartenant au même complexe de régulation. Les auteurs ont prouvé expérimentalement la haute régulation synergique induite par l'effet de la présence de ces deux régulateurs (FOXA1 et GATA3) ensemble, contrairement à la présence de chacun d'eux séparément.

5.8.4 Analyse des GRNs inférés

L'inférence de réseaux de régulation à large échelle est une tâche complexe et ce principalement à cause de la complexité du système biologique humain et à la difficulté d'identifier les mécanismes de régulations précis à partir des données d'expression de gènes humains.

Bien que nos résultats montrent que les GRNs inférés par H_SetNet contiennent un important nombre d'interactions supportées par plusieurs types de données (TFBS, ChIP, STRING) contrairement à d'autres méthodes d'inférence (Aracne et Genie3), la validité d'une interaction est encore difficile à estimer.

Pour cette raison, les méthodes d'inférence de réseau de régulation ont été appliquées avec succès sur des données de cancer chez l'humain pour découvrir de nouveaux *régulateurs masters* et pour comprendre leur rôle dans la progression du cancer [LRA+10].

Afin de déterminer la possibilité d'utiliser H_SetNet pour découvrir des régulateurs masters du cancer, nous avons analysé la liste des régulateurs ayant le plus grand nombre de gènes cibles. Parmi les 10 premiers régulateurs (*hub*) analysés, nous trouvons FOXA1, au deuxième rang. Particulièrement, FOXA1 est un régulateur de différenciation qui a été décrit comme étant associé à la progression de la tumeur de la vessie précédemment [DCC+12].

De plus, nous avons constaté que FOXM1 est classé sixième parmi les régulateurs qui ont le plus grand nombre de gènes cibles. FOXM1 est un régulateur master connu dans la progression de la tumeur et dans les métastases en général (voir [RP11] pour plus de détails). Ce dernier est découvert par l'application de méthodes d'inférence de réseaux de régulation dans d'autres types de cancer [LRA+10].

Par ailleurs, des études récentes ont validé expérimentalement le rôle du FOXM1 dans la prolifération des cellules cancéreuses en particulier dans le cancer de la vessie [LZK13].

L'ensemble de ces résultats confirme que H_SetNet est en mesure d'identifier correctement des régulateurs importants de cancer. Il a surtout l'avantage de pouvoir générer des solutions verifiées.

5.9 Conclusion

Dans ce chapitre, nous avons proposé une approche hybride H_SetNet d'inférence de réseaux de régulation. Cette approche se base sur un espace discret pour l'extraction de GRNs candidats et sur un espace numérique pour la sélection des GRNs les plus performants. H_SetNet a été validé sur deux types de données : Dream5 et données de cancer de vessie. Sur Dream5, la méthode H_SetNet a prouvé sa capacité de prédiction. Nous retenons plusieurs points forts : (i) H_SetNet a réduit considérablement le taux des faux positifs par rapport à la méthode discrete SetNet, tout en gardant l'important taux de vrais positifs. (ii) H_SetNet a démontré de bonnes performances en comparaison avec Aracne et avec la méthode gagnante du challenge Dream5, Genie3. (iii) H_SetNet est plus robuste que Aracne et Genie3 et continue à avoir de bonnes performances face à la contrainte de sous echantillonage.

Par ailleurs, sur les données humaines, l'étude expérimentale réalisée a prouvé que H_SetNet est la méthode la plus enrichie en interactions existantes de TFBS et ChIP, par rapport à Aracne et Genie3. En plus, une partie significative des relations cooperatives entre régulateurs prédites par H_SetNet est connue dans la base des interactions protéiques STRING. Notamment, la relation cooperative entre les deux régulateurs FOXA1 et GATA3 est expérimentalement prouvée parmi les complexes de régulation du Cancer du Sein, formant ainsi un avantage important de H_SetNet.

5.10 Bibliographie

Ce chapitre a été en partie publié dans :

- **Chebil, I**, Nicolle, R. and Santini, G. and Rouveirol, C. and Elati, M. Hybrid method inference for the construction of cooperative regulatory network in human *Bioinformatics and Biomedicine (BIBM), IEEE International Conference*, 2013.
- **Chebil, I**, Nicolle, R. and Santini, G. and Rouveirol, C. and Elati, M. Hybrid method inference for the construction of cooperative regulatory network in human *IEEE transactions on nanobioscience*, 2014.

Conclusion et perspectives

Bilan

Dans cette thèse, nous avons étudié le défi de l'inférence de réseaux de régulation génétique sur la base de données d'expression de gènes statiques. L'hypothèse de base est que les niveaux d'expression de gènes fournissent des informations/indications sur l'activité de régulation. Cependant, plusieurs contraintes compliquent la tâche d'inférence des régulations génétiques.

En effet, les données d'expression sont bruitées (*i.e.*, pouvant comporter des valeurs manquantes, être imprécises, etc.) et de haute dimension (*i.e.*, un grand nombre de gènes et un faible nombre d'échantillons). De plus, cette inférence doit aboutir à l'identification des facteurs de transcription interagissant en *coopération* afin d'activer ou de réprimer l'expression des gènes cibles. Comprendre ce fonctionnement coopératif des régulateurs représente l'un des enjeux majeurs de la modélisation du système de régulation sous-jacent.

Afin de modéliser le plus fidèlement possible les réseaux de régulations biologique et respecter ces contraintes, nous avons proposé dans cette thèse une nouvelle démarche d'inférence de réseaux de régulation basée sur la théorie d'ensemble.

Nous nous sommes appuyé sur Licorn, une méthode de fouille de données discrète développée au sein de notre équipe. Initialement, Licorn génère des GRNs candidats (*i.e.*, composés de complexes de régulation étiquetés) pour chaque gène cible. Ces GRNs sont évalués sur la base du score discret MAE, estimé à l'aide du programme de régulation RP. Ensuite, le GRN ayant le meilleur score est sélectionné. L'analyse de ces réseaux, sur les données Dream5, a montré que l'étape de la sélection qui se restreint au meilleur GRN détériore sensiblement le rappel de Licorn. De plus, cette sélection quasi-aléatoire d'un GRN parmi un ensemble de candidats portant le même score MAE dégrade sa précision (*pourquoi sélectionner ce réseau non pas un autre ?*). Par ailleurs, Licorn ne classe que les GRNs et ce selon le score MAE. Le classement des interactions qui les composent est absent, ce qui complique la phase de validation des interactions (*e.g.*, l'utilisation des données Dream5).

Notre approche consiste à exploiter les capacités de Licorn à identifier des réseaux locaux coopératifs tout en palliant à ces limitations. Tout d'abord, nous avons élaboré une méthodologie de sélection d'ensemble de GRNs à partir de l'ensemble des candidats extraits par Licorn. Nous avons basé le processus de sélection sur la théorie des méthodes d'ensemble afin de garantir que les GRNs sélectionnés soient à la fois précis et divers. Nous avons mis en place un algorithme glouton de sélection basé sur une fonction de score discrète qui respecte cette contrainte (critères de précision et de diversité). De plus, nous avons adopté deux mesures pour estimer la diversité entres GRNs (diversité de prédiction des GRNs et diversité de structure des GRNs).

Nous avons proposé un processus original de classement basé sur la régression linéaire répondant à deux critères : (i) le respect de la structure des réseaux lo-

caux et (ii) la quantification du poids de chaque interaction dans le réseau qui la compose ainsi que dans l'ensemble des réseaux sélectionnés. En d'autres termes, ce processus permet de prendre en considération non seulement le poids d'un régulateur dans un GRN mais aussi (et surtout) le poids d'un GRN dans l'ensemble des GRNs inférés.

Cette méthode a été implémentée dans un outil appelé SelectNet et expérimentée sur les données du challenge Dream5. Les résultats obtenus ont permis de répondre aux limites de Licorn. D'abord, un grand nombre d'interactions TP composent les GRNs sélectionnés par SelectNet. Ensuite, ces interactions ont été bien classées. Néanmoins, cette expérimentation a soulevé une contradiction : nous avons constaté une dégradation de performance suite à l'introduction de diversité dans le processus de sélection ce qui est contradictoire avec la théorie des méthodes d'ensembles qui stipule qu'un ensemble doit être divers et précis à la fois pour garantir une meilleure performance.

Afin de respecter la définition de l'approche ensembliste, nous avons proposé une extension de SelectNet, appelée SetNet, dans laquelle nous avons opté pour l'adoption du bagging afin d'introduire de l'aléa dans les données d'apprentissage. L'utilisation de cette technique a conduit à des changements dans les modèles appris permettant ainsi de découvrir de nouveaux GRNs candidats (*i.e.*, des nouvelles interactions).

Les expérimentations menées sur les données Dream5 nous ont permis de faire deux constatations :

- Une forte amélioration de performance : les réseaux inférées par SetNet sont largement plus riches en interactions TP que SelectNet. De plus, SetNet est capable d'inférer un grand nombre d'interactions TF non identifiées par deux algorithmes de l'état d'art, à savoir Genie3 —la méthode gagnante du challenge Dream5— et Aracne.
- Un grand nombre d'interactions FP : l'analyse des réseaux inférés a révélé qu'un grand nombre d'interaction FP sont aussi sélectionnées. Ce constat met en doute la capacité de la fonction discrète de sélection — basée sur le programme de régulation — à discriminer les vraies interactions des mauvaises générant ainsi un nombre important de FP.

Pour pallier à cette problématique, nous avons opté pour une nouvelle stratégie de sélection qui abandonne le programme de régulation local discret peu précis pour une approche numérique. Nous avons proposé donc une méthode hybride appelée H_SetNet qui génère —comme est le cas pour SetNet— des GRNs candidats suivant un modèle local discret de régulation, puis sélectionne les GRNs les plus précis —contrairement à SetNet— selon une méthode de sélection numérique. Deux techniques numériques de sélection ont été expérimentées, une technique supervisée et une non supervisée.

L'évaluation de ces techniques a prouvé que la technique supervisée donne de meilleurs résultats. De ce fait, cette technique a été incorporé dans H_SetNet.

Nous avons évalué H_SetNet sur les données Dream5 et nous l'avons comparé en premier lieu à SetNet, puis à Genie3 et à Aracne. Les résultats obtenus ont montré que :

- H_SetNet est capable de discriminer les bonnes interactions des mauvaises contrairement à SetNet, et ce en discriminant un nombre très important des FP.
- H_SetNet est la méthode qui infère le moins d'interactions FP en comparaison à Genie3 et Aracne.

Une autre question importante que nous avons abordée dans cette thèse est la capacité de H_SetNet à garder de bonnes performances face à la contrainte de la haute dimension des données (sous échantillonnage). Pour cela, nous avons simulé cette situation en sélectionnant des sous-échantillons aléatoires de petites tailles des données Dream5 formant ainsi des sous-ensembles de données. Dans la phase de test, nous avons appliqué H_SetNet ainsi qu'Aracne et Genie3 sur ces sous-ensembles afin de vérifier leur sensibilité à la contrainte de sous échantillonnage. Les résultats obtenus montrent que non seulement H_SetNet garde de bonnes performances face à la contrainte de sous échantillonnage mais c'est la méthode la plus robuste au faible nombre d'échantillons par rapport à Aracne et Genie3.

La finalité de notre travail étant de pouvoir valider notre approche H_SetNet sur des données humaines, nous avons tenté de répondre aux questions suivantes :

- H_SetNet est-il capable d'identifier des relations de régulations humaines ?
- Les relations coopératives inférées par H_SetNet ont-elles une signification biologique ?

Afin de répondre à ces questions, nous avons collaboré avec Mohamed Elati et Rémy Nicolle de l'équipe MEGA de l'institut de Biologie Systémique et Synthétique (iSSB). Nous avons réalisé une étude expérimentale de H_SetNet sur des données du cancer de la vessie. Contrairement à l'évaluation des méthodes d'inférence sur Dream5, les relations de régulations réelles pour les données humaines ne sont pas connues. Pour cette raison, nous avons utilisé plusieurs types de données auxiliaires (*i.e.*, TFBS, ChIP et STRING) afin d'évaluer la pertinence des prédictions. Ces données sont sous la forme de listes d'interactions possibles extraites de la base TRANSFAC®. Les résultats obtenus ont prouvé que H_SetNet est la méthode qui infère le plus grand nombre d'interactions existantes de TFBS et ChIP, en comparaison à Aracne —l'algorithme le plus utilisé dans les applications de biologie des systèmes du cancer— et à Genie3. En plus, une partie significative des relations coopératives entre régulateurs prédites par H_SetNet existe dans la base des interactions protéiques STRING.

Tous ces résultats positifs — sur les données Dream5 et les données humaines — sont prometteurs et prouvent que la méthodologie que nous avons proposée est bien fondée.

Perspectives

Étant donnée la richesse du domaine, les perspectives de ce travail couvrent différentes disciplines. En effet, et comme nous l'avons évoqué dans l'introduction de cette thèse, la modélisation en biologie requiert une approche multi-disciplinaires et demande ainsi des connaissances dans des domaines scientifiques variés.

Meta-ensemble Le premier constat que nous avons tiré de la comparaison de plusieurs méthodes d'inférence est la grande variation des réseaux inférés. En effet, les différents diagrammes de Venn que nous avons présenté (e.g. diagrammes des sections 4.3.2 et 5.6.2) montrent que chaque méthode a un nombre d'interactions réelles (TP) qu'elle est la seule à pouvoir inférer. Ainsi, on peut imaginer une approche qui combine les résultats des différentes méthodes afin de maximiser le nombre d'interactions prédites. Cette méthode – qu'on peut appeler *meta-ensemble* – étend le concept d'ensemble en exploitant la différence

de prédiction entre les modèles qui peut être assimilée à l'introduction de diversité.

Une approche similaire a été proposée par Vignes M. et al. [VVA+11]. Cette dernière combine les résultats de plusieurs méthodes d'inférence dans un système appelé *méta-analyse*. Spécifiquement, les auteurs ont cherché un consensus entre plusieurs modèles de prédiction (*i.e.*, les réseaux bayésiens, les régressions LASSO) et ont montré que leur approche améliore sensiblement les performances de prédiction sur les données de Dream5.

Réseau global L'approche considérée dans cette thèse extrait des réseaux locaux de régulation. L'étendre à un réseau global peut contribuer à une meilleure compréhension du fonctionnement de la régulation. Rappelons tout d'abord que le choix d'extraire des réseaux locaux est motivé par la faible complexité des calculs nécessaires pour effectuer cette tâche. En effet, dans ce cas de figure, la recherche des réseaux locaux peut se faire d'une manière parallèle réduisant considérablement le temps de calcul, ce qui permet d'analyser des données à grande échelle (e.g. données humaine). L'établissement d'un tel modèle global à partir des motifs locaux nécessite à la fois de combiner des motifs de façon efficace mais surtout de gérer une éventuelle redondance ou des conflits entre ces motifs.

Expérimentation biologique Le but principal de notre approche d'inférence est de pouvoir aider à orienter les futures études expérimentales. Il est ainsi essentiel de pouvoir tester notre méthode sur des données réelles et de pouvoir valider les résultats avec les biologistes. La construction du modèle d'inférence pour les réseaux de régulations n'est pas idéale (*i.e.*une approximation), il est donc nécessaire de critiquer, améliorer, parfois simplifier ou au contraire complexifier le modèle suivant les nouvelles découvertes biologiques. Nos divers expérimentations sur les jeux de données Dream5 ont permis d'avoir une première validation. Nous avons par la suite étendu cette validation à des données humaines afin de confirmer la validité de notre approche. Il faut maintenant étendre ces expérimentations à d'autres jeux données ainsi que valider les réseaux obtenus avec les biologistes.

Troisième partie

Annexes

Annexe A

Quelques notions de la génomique et de la post-génomique

La *génomique* est la science dont le but est de déterminer le contenu génétique de tout le génome et de comprendre la régulation des gènes. Depuis les années 1990, on a vu la quantité de données biologiques disponibles augmenter exponentiellement grâce à des avancées à la fois en biologie et en bioinformatique (la post-génomique). Dans cette annexe, il ne s'agit pas d'entrer dans les détails biologiques mais de montrer notions importantes pour la compréhension de la problématique abordée dans cette thèse. Beaucoup de notions utilisées se trouvent dans tout livre d'initiation à la biologie moléculaire et/ou la (post-)génomique [DK02].

A.1 Éléments de la biologie moléculaire

Nous présentons dans cette section les éléments essentiels permettant d'appréhender les mécanismes de régulation de la transcription.

La cellule

La *cellule* est une unité élémentaire structurale et fonctionnelle constituant tout ou partie d'un organisme vivant. La cellule est constituée d'un système, le métabolisme, défini comme l'ensemble des transformations moléculaires et des transferts d'énergie qui se déroulent de manière ininterrompue dans la cellule ou l'organisme vivant. C'est un processus ordonné (voir Figure A.1), qui fait intervenir des processus de dégradation et de synthèse organique. Pour mieux appréhender la complexité d'une cellule, on peut décomposer son métabolisme selon différents concepts :

- Les voies métaboliques : c'est la décomposition du métabolisme en sous-systèmes plus intelligibles, comme par exemple la dégradation du glucose.
- La transduction du signal : c'est un processus par lequel la cellule convertit un type de signal ou stimulus d'origine externe à la cellule en un autre. Elle résulte en général en l'activation ou l'inactivation de fonctions cellulaires.
- La régulation des gènes : c'est le terme général pour la modulation de l'expression des gènes, c'est-à-dire le contrôle cellulaire de la quantité et du moment de la présence du produit d'un gène.

La synthèse des protéines

L'expression des gènes codant pour des protéines consiste en une succession de deux grandes étapes qui vont permettre de produire, à partir de l'ADN, des protéines (voir

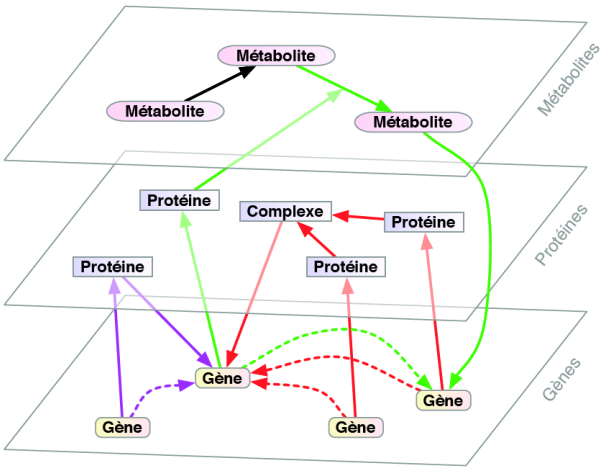


FIGURE A.1 – Vue simplifiée du réseau macromoléculaire formant la cellule, ne reprenant que les classes de macromolécules les plus utilisées : gènes, protéines et métabolites, regroupées par niveaux ou espaces. Figure adaptée de [BdlFM02]

Figure A.2) : la transcription et la traduction. Lors de la transcription, une molécule intermédiaire – l'ARN messager (ARNm) – est synthétisée dans le noyau en utilisant la séquence d'ADN d'un gène comme modèle. Puis l'ARN messager subit une phase de maturation et d'épissage afin de produire un ARN mature qui pourra être traduit en protéine. Lors de cette phase, les régions non codantes de l'ARN, nommées introns, sont excisées pour ne conserver que les portions codantes, appelées exons. L'ARN messager, ainsi obtenu, est ensuite transporté à l'extérieur du noyau pour être traduit en protéine. Lors de cette deuxième étape de traduction, les triplets de nucléotides de l'ARN sont traduits en acides aminés et assemblés pour former une protéine.

Régulation de l'expression

La synthèse des protéines est contrôlée en fonction du contexte dans lequel se trouve la cellule. Cette régulation, ou modulation de l'expression, intervient à tous les niveaux de la synthèse des protéines. En particulier, chez les eucaryotes, elle peut intervenir au niveau de :

- l'activation de la structure chromatinienne,
- l'initiation de la transcription,
- l'étape de maturation de l'ARN,
- l'étape de transport de l'ARN en dehors du noyau,
- l'étape de traduction,
- la dégradation des objets (ARN messager et protéines)

Lorsqu'un gène est actif, c'est-à-dire lorsque sa structure chromatinienne est présente sous forme non condensée, une part importante de régulation a lieu au niveau transcriptionnel et plus particulièrement lors de la phase d'initiation. Nous concentrons cette introduction sur ce niveau de régulation pour deux raisons. D'une part, le contrôle de la traduction de l'ADN en ARN conditionne les étapes suivantes de l'expression. D'autre

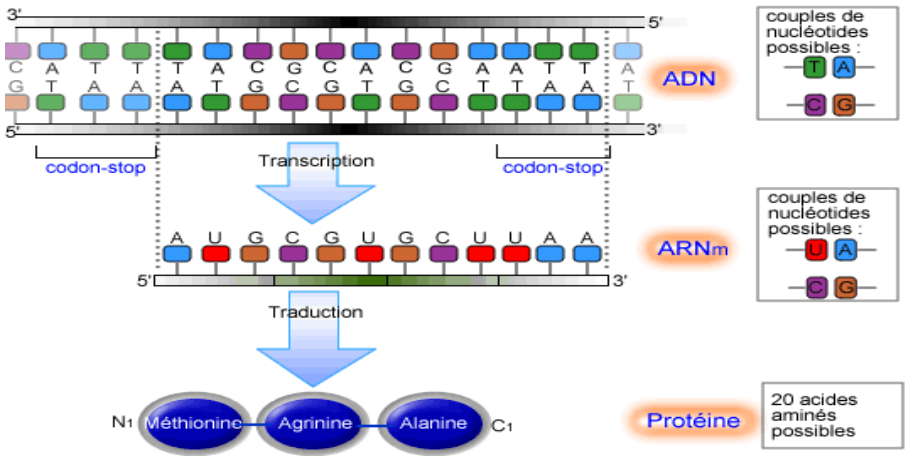


FIGURE A.2 – Schéma de la synthèse des protéines.

part, c'est sur ce mode de régulation qu'il existe actuellement le plus de connaissances, de techniques expérimentales et de données disponibles.

Facteurs de transcription

L'arrimage de l'ARN polymérase sur l'ADN nécessite la présence préalable de protéines particulières appelées facteurs généraux de transcription. Dans un premier temps, ces facteurs se fixent de manière séquentielle en amont du gène à transcrire dans une région appelée promoteur en formant un complexe. Ensuite ce complexe va recruter l'ARN polymérase et la positionner au niveau d'un site localisé en amont du gène, appelé site de fixation. La transcription peut alors commencer. En association des facteurs généraux, d'autres facteurs, les facteurs de transcription spécifiques, vont intervenir et influer de manière positive ou négative sur la transcription. Il convient ici de distinguer précisément les facteurs de transcription généraux ou basaux, dont la présence est requise pour initier la transcription, des facteurs spécifiques qui possèdent une action régulatrice sur l'expression propre à chaque gène.

D'un point de vue biochimique, les facteurs de transcription sont des protéines possédant des domaines de fixation à l'ADN et des domaines d'activation de la transcription. Les domaines de fixation vont leur permettre de se fixer à l'ADN sur de courtes séquences spécifiques : les sites de fixation de facteurs de transcription.

Dans l'assemblage ADN-protéine, on appelle également élément trans-régulateur le facteur de transcription, et élément cis-régulateur le site de fixation nucléique reconnu. Les sites peuvent correspondre aux séquences de fixation de plusieurs facteurs ce qui dans ce cas définit un module cis-régulateur. L'interaction ADN-protéine repose sur une forme de complémentarité entre la composition nucléique du site de fixation et le site actif de la protéine régulatrice. Les sites sur lesquels se fixent les facteurs de transcription sont de courts segments d'ADN (une dizaine de nucléotides) qui présentent une forte variabilité au niveau de leurs séquences nucléiques.

Transcription

Certaines régions des chromosomes correspondent aux gènes. Lorsqu'un gène est exprimé, sa séquence est recopiée, c'est le mécanisme de la transcription, en un polymère d'acides ribonucléiques ou ARN. Les acides ribonucléiques sont légèrement différents des acides désoxyribonucléiques : ils possèdent un groupement hydroxyle (OH) en plus dans la partie non variable des nucléotides, et la thymine (T) est remplacée par l'Uracyle (U). L'alphabet des ARN est donc A,C,G,U. La séquence d'ARN correspondant à un gène destiné à produire une protéine correspond à un message contenant l'information pour produire une protéine et est appelée ARN messager ou ARNm. La technologie des puces à ADN (microarray en anglais) [SSDB95] permet de mesurer la quantité de chacun des ARNm présents dans une cellule. Cette information s'appelle le transcriptome et elle reflète le niveau d'expression de l'ensemble des gènes d'une cellule à un instant donné dans certaines conditions.

Le transcriptome d'une cellule est une information utile à l'étude de la régulation des gènes. En effet, en faisant varier les conditions expérimentales de culture de cellules et en mesurant le niveau d'expression de tous les gènes dans ces différentes conditions, on peut « voir » les gènes qui sont activés ou désactivés afin de répondre aux changements environnementaux. Précisons que la régulation de l'expression des gènes peut également s'effectuer en aval de la transcription et dans ce cas, on parle de régulation post-transcriptionnelle (voir [McC98] pour plus de détails). Les données obtenues par puces à ADN ne reflètent donc pas fidèlement le niveau d'expression de tous les gènes. Lorsque les conditions expérimentales sont modifiées, on observe des groupes de gènes qui « réagissent » de manière coordonnée et on en déduit une relation de co-régulation entre ces gènes. On suppose alors qu'ils participent à un même rôle cellulaire comme, par exemple, l'activation d'une certaine voie métabolique.

A.2 La génomique

Le premier génome complet est apparu en 1977, celui du bactériophage phi X174 [SAB+77], un virus. Le matériel génétique d'un organisme est contenu dans son ou ses chromosomes, il s'agit de molécules d'acide désoxyribonucléique ou ADN. Ces molécules sont des polymères de petites molécules de base, les acides nucléiques ou nucléotides, qui diffèrent entre elles par une partie appelée base azotée. Il existe quatre types de bases azotées dans l'ADN, généralement notées A (adénine), T (thymine), C (cytosine) et G (guanine). La molécule d'ADN est constituée d'un enchaînement linéaire de nucléotides, appelé brin d'ADN. Dans la plupart des organismes, l'ADN contenu dans les chromosomes est composé de deux brins appariés, appelés brins antiparallèles. Formellement, il suffit de représenter un seul brin sous la forme d'un mot sur l'alphabet $\{A, C, G, T\}$. Le génome d'un organisme correspond donc aux mots représentant la séquence de chacun de ses chromosomes (voir Figure A.3).

Il existe principalement trois grandes banques de données formant l'International Nucleotide Sequence Databases (INSD) dédiées à la mise à disposition des séquences des gènes et des génomes (EMBL en Europe, GenBank aux États-Unis et DDBJ au Japon). Ces banques de données ont une importance capitale car elles offrent un accès public aux séquences connues.

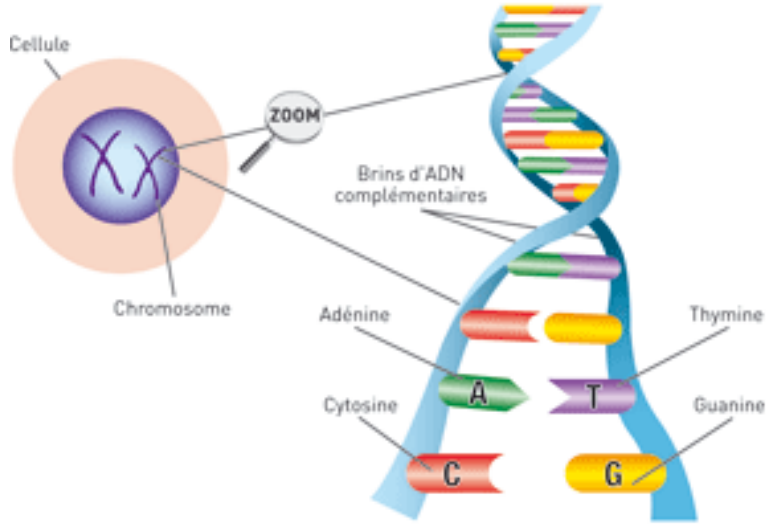


FIGURE A.3 – Représentations schématiques d'un segment de molécule d'ADN.

La connaissance de génomes complets est essentielle car elle permet d'accéder potentiellement à la totalité des gènes d'une espèce et d'étudier comment les gènes sont organisés sur les chromosomes. Entre autres, ces connaissances permettent :

- les études globales sur l'ensemble des gènes d'un organisme,
- les études en génomique comparative et évolution des espèces : on peut comparer les génomes d'espèces plus ou moins éloignées en terme d'évolution, les duplications, fusions ou pertes de gènes,
- la conception de puces à ADN,
- l'accès au protéome complet de l'organisme.

A.3 Méthodes de la post-génomique

Depuis l'avènement de la (post-)génomique, une classe de techniques permettent d'acquérir des données moléculaires de manière massivement parallèle a émergé. Nous trouvons des techniques qui permettent de mesurer la quantité d'ARNm au sein de la cellule, telles que les puces à ADN [SSDB95], et d'autres qui permettent de prédire des liaisons de facteurs de transcription sur la région régulatrice de leurs gènes cibles, telle que la technique de ChIP-Chip [HS02]. Dans cette section, nous décrivons ces deux principales techniques.

Technologie des puces à ADN

La puce à ADN [SSDB95] est l'outil de post-génomique le plus répandu. Il permet de mesurer les niveaux d'expression de plusieurs milliers de gènes simultanément et offre ainsi la possibilité d'étudier des génomes entiers, comme par exemple le génome humain qui compte environ 30 000 gènes. Le succès de la technologie a entraîné, depuis le début des

années 2000, un essor considérable de leur utilisation pour étudier différents organismes et phénomènes biologiques. La mesure à grande échelle de l'expression génétique est motivée, entre autres, par l'hypothèse que l'état fonctionnel d'un organisme est en grande partie décrit par la quantité de chaque ARNm présent dans la cellule à un moment donné. Le transcriptome d'une cellule est une information utile à l'étude de la régulation des gènes. En effet, en faisant varier les conditions expérimentales de culture de cellules et en mesurant le niveau d'expression de tous les gènes dans ces différentes conditions, nous pouvons observer les gènes qui sont activés ou désactivés afin de répondre aux changements environnementaux.

L'idée conceptuelle de la puce à ADN est très simple. Il s'agit de greffer sur une surface de quelques centimètres carrés des fragments synthétiques d'ADN (les sondes) représentatifs de chacun des gènes que l'on souhaite étudier et espacés de quelques micromètres. Ce micro-dispositif est ensuite mis au contact des acides nucléiques à analyser (voir Figure pour un aperçu de la techniqueA.4).

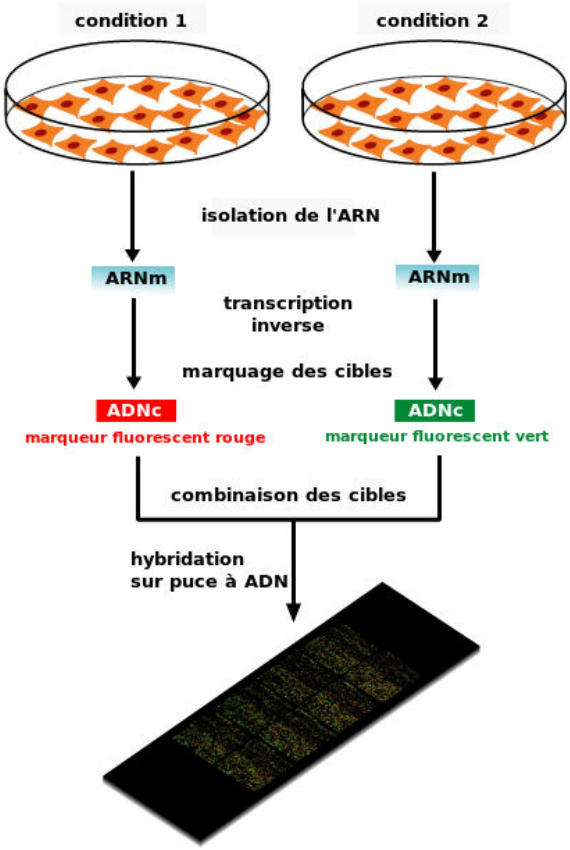


FIGURE A.4 – Schéma récapitulatif du principe des puces à ADN. Source : *http ://fr.wikipedia.org/wiki/Puce_à_ADN*. Pour chaque ARN étudié, un ensemble de cibles est préparé par amplification et marquage. Ces cibles sont ensuite hybridées avec les sondes, complémentaires, immobilisées en spots sur une lame. On quantifie ensuite le signal associé à chaque spot

Les deux technologies dominantes sont les puces dites *spottées* par un dépôt robotisé de produits de PCR ou de longs fragments oligonucléiques (*spotted microarrays*) et les

puces à oligonucléotides synthétisés in situ (*GeneChips* de la société Affymetrix) :
La méthode de fabrication des puces « spottées » a été développée par l'équipe de P. Brown à l'université de Stanford, aux États-Unis. Elle est aujourd'hui bien établie et de nombreuses plate-formes de production sont implantées dans les laboratoires académiques. Des solutions d'ADN sont préparées soit par amplification PCR à partir du génome ou de banques d'ADN complémentaires, soit par synthèse d'oligonucléotides longs (30-70 mers). Des micro-gouttelettes de ces solutions sont ensuite déposées par un robot, selon une matrice d'emplacements définis, sur une lame de verre traitée par un revêtement chimique qui permet de fixer l'ADN. En général, chaque spot de la matrice correspond à un gène donné. Les robots nécessaires à la fabrication de ces puces étaient construits à l'origine de manière artisanale dans chaque laboratoire selon le modèle conçu par DeRisi [DIB97].
Les puces à oligonucléotides synthétisés in situ par photolithographie [La96] (*GeneChips* de la société Affymetrix) ne peuvent être produites que par des sociétés industrielles spécialisées, mais elles sont également de plus en plus utilisées et elles bénéficient désormais d'une importante diversification, d'une certaine baisse des prix et d'un contrôle de qualité accru. Une contrainte souvent posée par l'utilisation de ces puces est qu'elle nécessite en général l'emploi de méthodes et d'équipements imposés par le fournisseur (type de lecteurs, de logiciels d'analyse) et que les licences de propriété industrielle ne permettent pas l'accès à certaines informations.

Quel que soit le type de puces, le succès de la technologie a entraîné, depuis le début des années 2000, un élargissement considérable du choix des équipements et des protocoles expérimentaux, aussi bien pour la fabrication des lames que pour l'amélioration des conditions de manipulation en vue d'optimiser la sensibilité, la spécificité et la reproductibilité de la méthode. Les études exploitant l'utilisation des puces à ADN se multiplient rapidement dans des domaines d'application variés. Le niveau d'expression de chacun des gènes est représenté sur la puce par un à quelques spots de sondes pour ce qui concerne les puces de faible et moyenne densité. Après traitement de l'image, le signal analysé est en général, la moyenne (ou le médiane) des pixels composant chaque plot, moyenne à laquelle on retranche souvent une valeur de bruit de fond estimée dans la zone périphérique du plot. Les puces à haute densité, commercialisées par la société Affymetrix, ont une structure très particulière. Pour chaque gène, une série de dix à vingt sondes, répartie sur toute la séquence du gène, est représentée sur la lame.

Technique de ChIP-Chip

Le ChIP-Chip [HS02, LRR+02] est une technique à grande échelle qui permet d'identifier l'ensemble des sites de fixation d'un facteur de transcription sur la chromatine (voir Figure A.5).

Le ChIP (Chromatin ImmunoPrecipitation) consiste à immunoprécipiter la chromatine au moyen d'un anticorps dirigé contre le facteur de transcription étudié. L'existence d'un anticorps spécifique est un point clé dans la technologie de ChIP. L'avantage de cette technique est qu'elle permet d'observer les interactions dans les conditions physiologiques, sans créer de perturbation pouvant modifier l'expression des gènes. Suite à une purification, il est possible de récupérer les régions d'ADN génomiquessur lesquelles était fixé le facteur de transcription. Ces ADN sont ensuite hybridés, après marquage et amplification, sur une puce pangénomique.

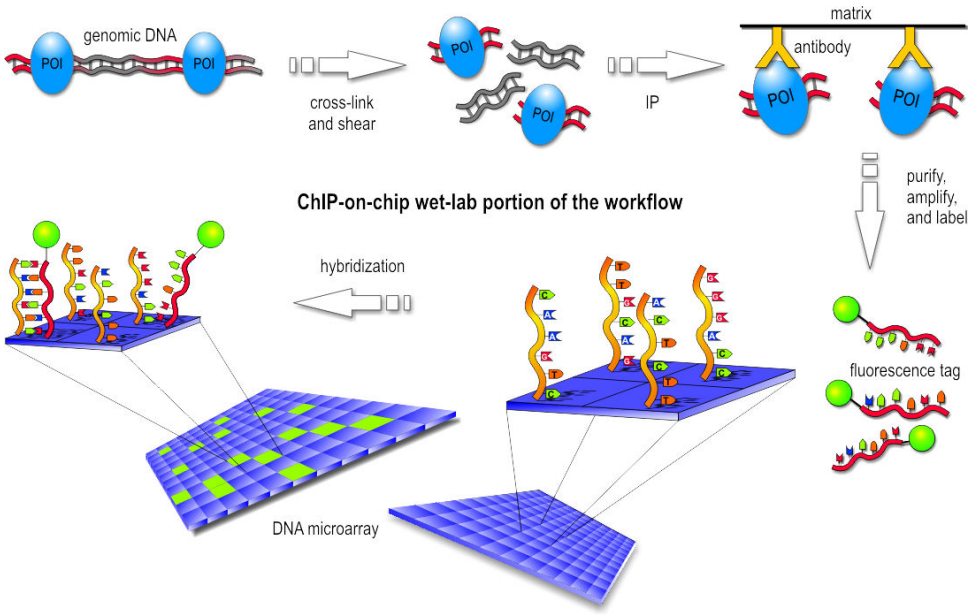


FIGURE A.5 – Technique de *ChIP-Chip*

Après hybridation, on obtient deux groupes de séquences, celles dites positives en ChIP qui possèdent un site de fixation au facteur de transcription, et celles dites négatives en ChIP, où le facteur de transcription n'a pas pu se fixer. Il est alors possible de rechercher, ou de découvrir, les motifs reconnus par un facteur de transcription donné. Cette technique, a été utilisée d'abord chez la levure **??**, et ensuite chez l'homme [CMS+06, LGLM+06].

A partir des sites de fixation du facteur de transcription étudié en ChIP-Chip, il est possible de découvrir des motifs caractérisant d'autresfacteurs de transcription à proximité de ce dernier, interagissant avec lui au cours de la régulation transcriptionnelle. Si ces motifs correspondent à des facteurs de transcription connus, pour lesquels il existe des anticorps spécifiques, il sera possible de réaliser de nouveaux ChIP-Chip, afin de valider le réseau. Cela pourra également permettre de découvrir de nouveaux facteurs à leur voisinage. L'intégration des données d'expression et du ChIP-chip peut donc permettre de construire des réseaux de régulation transcriptionnelle, où il sera possible de visualiser l'ensemble des facteurs de transcription intervenant dans le mécanisme de régulation, ainsi que leurs cibles. De tels réseaux pourront permettre la recherche de cibles thérapeutiques afin de rétablir une régulation transcriptionnelle non pathologique.

Annexe B

Évaluation de l'apprentissage

Une des difficultés de l'inférence de réseaux de régulation est l'évaluation des résultats obtenus, car il n'y a pas un critère objectif pour définir ce qu'est une "bonne" interaction entre deux gènes [WHRG03]. Cependant, le réseau de régulation une fois appris, peut être vu comme un classifieur, il s'agit de prédire le niveau d'expression (classe) du gène cible à partir de ces régulateurs. Ce qui permet de tester le pouvoir en généralisation de la méthode d'inférence. Cette annexe présente un panorama sur les différentes techniques d'évaluation d'apprentissage avec des pointeurs vers leurs utilisations dans le cadre de l'inférence de réseaux de régulation.

B.1 L'erreur en généralisation

L'évaluation de la performance d'un algorithme d'apprentissage est un point crucial. En effet, c'est à partir de cette évaluation que l'on peut comparer, choisir ou valider les différentes méthodes. On cherche à évaluer, en particulier, l'erreur en généralisation des modèles prédictifs que nous avons construits, c'est-à-dire le taux d'erreur de prédiction de la sortie d'un ensemble d'exemples indépendants de ceux utilisés pendant l'apprentissage. Pour cela on définit tout d'abord une fonction qui mesure l'erreur entre la sortie du modèle et la sortie réelle.

La performance sur les données d'apprentissage est intrinsèquement optimiste, mais, en outre, son comportement n'est pas forcément un bon indicateur de la vraie performance. Un phénomène classique, présenté dans [CM02] (voir Figure B.1), est que l'erreur empirique diminue au fur et à mesure que le système prend en compte davantage d'informations tandis que l'erreur réelle, d'abord décroissante, se met à augmenter après un certain stade. Ce phénomène est appelé *sur-apprentissage* (*over-fitting*). Une meilleure approche serait de calculer l'erreur en classification sur un ensemble de test contenant des exemples autres que ceux utilisés lors de l'apprentissage. Si on dispose d'un nombre d'exemples suffisant, l'erreur sur l'ensemble de test sera une bonne approximation de l'erreur réelle. Malheureusement les jeux de données qui nous intéressent (données de transcriptome) contiennent peu d'exemples (quelques dizaines). Il faut donc recourir à d'autres méthodes telles que la validation croisée ou le *leave-one-out*, etc. Ces différentes méthodes sont décrites dans la suite de cette section.

B.1.1 Utilisation d'un échantillon de test

La méthode la plus simple pour estimer la qualité objective d'une hypothèse d'apprentissage h est de diviser l'ensemble des exemples en deux ensembles indépendants : le premier, noté $\mathcal{A}$, est utilisé pour l'apprentissage de h et le second, noté $\mathcal{T}$, sert à mesurer sa qualité. Ce second ensemble est appelé *échantillon (ou ensemble d'exemples) de test.*

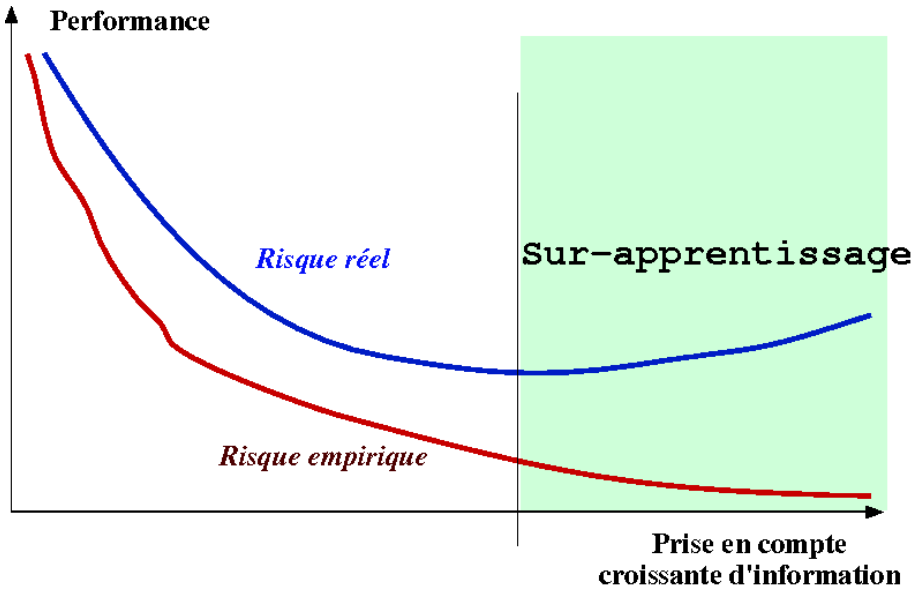


FIGURE B.1 – Illustration du phénomène de sur-apprentissage. Tandis que l'erreur empirique continue de diminuer au fur et à mesure de la prise en compte d'information, l'erreur réelle qui diminuait également dans un premier temps, commence à réaugmenter après un certain stade. Il n'y a alors plus de corrélation entre l'erreur empirique et l'erreur réelle. La figure est extraite de [CM02]

On a $\mathcal{S} = \mathcal{A} \cup \mathcal{T}$ et $\mathcal{A} \cap \mathcal{T} = \emptyset$. Comme nous allons le voir, la mesure des erreurs commises par h sur l'ensemble de test $\mathcal{T}$ est une estimation de l'erreur réelle de h. Cette estimation se note :

$$\widehat{E}_{Réelle}(h)$$

Dans le cas de l'apprentissage d'une règle de classification, l'estimation de cette erreur se fait à l'aide d'une matrice de confusion. La *matrice de confusion* $M(i,j)$ d'une règle de classification h est une matrice $C \times C$ dont l'élément générique donne le nombre d'exemples de l'ensemble de test $\mathcal{T}$ de la classe i qui ont été classés dans la classe j.

Dans le cas d'une classification binaire, la matrice de confusion est donc de la forme :

	1	**0**
1	Vrais positifs	Faux positifs
0	Faux négatifs	Vrais négatifs

Si toutes les erreurs sont considérées comme également graves, la somme des termes non diagonaux de M, divisée par la taille t de l'ensemble de test, est une estimation $\widehat{E}_{Réelle}(h)$ sur $\mathcal{T}$ du risque réel de h.

$$\widehat{E}_{Réelle}(h) = \frac{\Sigma_{i \neq j} M(i,j)}{t}$$

En notant t_{err} le nombre d'objets de l'ensemble de test mal classé, on a donc :

$$\widehat{E}_{Réelle}(h) = \frac{t_{err}}{t}$$

B.1.2 Estimation par validation croisée

L'idée de la validation croisée (*N-fold cross-validation*) consiste à :
- Diviser les données d'apprentissage $\mathcal{S}$ en N sous-échantillons de tailles égales ;
- Retenir l'un de ces échantillons, disons de numéro i, pour le test et apprendre sur les $N-1$ autres ;
- Mesurer le taux d'erreur empirique $\widehat{E}^i_{Réelle}(h)$ sur l'échantillon i ;
- Recommencer n fois en faisant varier l'échantillon i de 1 à N.

L'erreur estimée finale est donnée par la moyenne des erreurs mesurées :

$$\widehat{E}_{Réelle}(h) \;=\; \frac{1}{N}\sum_{i=1}^{N}\widehat{E}^i_{Réelle}(h)$$

Cette procédure fournit une estimation non biaisée du taux d'erreur réel. Il est courant de prendre pour N des valeurs comprises entre 5 et 10. De cette manière, on peut utiliser une grande partie des exemples pour l'apprentissage tout en obtenant une mesure précise du taux d'erreur réel. En contre partie, il faut réaliser la procédure d'apprentissage N fois. La question se pose cependant de savoir quelle hypothèse apprise on doit finalement utiliser. Il est en effet probable que chaque hypothèse apprise dépende de l'échantillon i utilisé pour l'apprentissage et que l'on obtienne donc N hypothèses différentes. Le mieux est alors de refaire un apprentissage sur l'ensemble total $\mathcal{S}$. La précision sera bonne et l'estimation du taux d'erreur est connue par les N apprentissages faits précédemment.

Lorsque les données disponibles sont très peu nombreuses, il est possible de pousser à l'extrême la méthode de validation croisée en prenant pour N le nombre total d'exemples disponibles. Dans ce cas, on ne retient, à chaque fois, qu'un seul exemple pour le test, et on répète l'apprentissage N fois pour tous les autres exemples d'apprentissage. Cette méthode est connue sous le nom de *leave-one-out*.

B.2 L'estimation par les courbes ROC et PR

Jusqu'ici, nous avons essentiellement décrit des méthodes d'évaluation des performances ne prenant en compte qu'un nombre : l'estimation de l'erreur réelle. Cependant, il peut être utile d'être plus fin dans l'évaluation des performances et de prendre en compte non seulement un taux d'erreur, mais aussi les taux de faux positifs et de faux négatifs. Souvent, en effet, le coût de mauvaise classification n'est pas symétrique et l'on peut préférer avoir un taux d'erreur un peu moins bon si cela permet de réduire le type d'erreur le plus coûteux.

La courbe *ROC* (de l'anglais *Receiver Operating Characteristic*) et la courbe PR (de l'anglais *Precision Recall*) permettent de régler ce compromis [Bra97].

Plus en détail, la courbe ROC [Bra97] (voir Figure B.2) est une représentation graphique de la relation existante entre la sensibilité (*taux de vrais positifs TPR*) et la spécificité (*taux de faux positifs FPR*) d'un modèle, calculée pour toutes les valeurs seuils possibles.

En effet, la sensibilité d'un modèle est estimé par le nombre des vrais positifs (TP) par rapport à tous les positifs (vrais positifs et faux négatifs FN) quelque soit l'état de leur prédiction :

$$TPR = TP/(TP+FN) \tag{B.1}$$

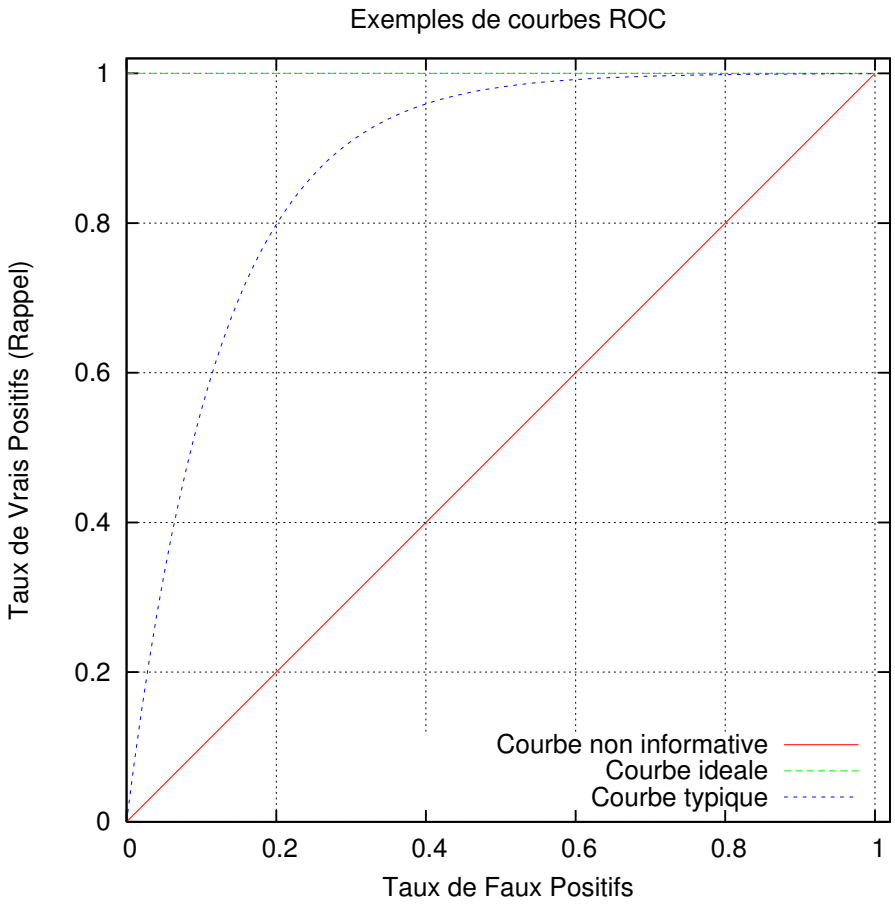


FIGURE B.2 – Exemples de courbes ROC pour l'estimation de la performance d'un algorithme d'inférence de réseaux de régulation.

Quant à la spécificité, elle est estimée par le nombre des vrais négatifs (TN) par rapport à tous les négatifs (vrais négatifs et faux positifs FP) quelque soit l'état de leur prédiction :

$$FPR = TN/(TN + FP) \tag{B.2}$$

Quant à la courbe PR (*Precision Recall*), c'est une représentation graphique de la relation entre la précision et le rappel d'un modèle, calculée pour toutes les valeurs seuils possibles. La précision représente le pourcentage des prédictions correctes alors que le rappel représente le pourcentage des prédictions correctes inférées.

$$Precision = TP/(TP + FP) \tag{B.3}$$

$$Recall = TP/(TP + FN) \tag{B.4}$$

Notons que l'air sous les courbes ROC et PR (respectivement $AUROC$ et $AUPR$) est un indicateur de performance d'un algorithme, une valeur plus élevée indique une meilleure performance.

Toutefois, pour évaluer la performance des techniques d'inférence de réseaux de régulation en utilisant la courbe ROC et la courbe PR, le vrai réseau sous-jacent doit être

connu. Or, nos connaissances biologiques des interactions génétiques sont très incomplètes et par la suite les vrais positifs et les faux positifs ne peuvent pas toujours être évalués.

Pour contourner cette limite, une nouvelle tendance de test sur des données synthétiques d'expression à partir d'un réseau artificiel [SJH02] est adoptée. Ces données permettent d'effectuer une étude systématique de la performance d'un algorithme (*e.g.*, paramètres de l'algorithme, étude de la résistance au bruit, taille de l'ensemble d'apprentissage), et permettent de réaliser une comparaison de performance entres plusieurs algorithmes.

Nous décrivons, dans ce qui suit, les données les mieux adaptées et les plus utilisées pour l'évaluation des méthodes d'inférence de réseaux de régulation.

B.3 Les challenges DREAM

DREAM [1] est un dialogue pour l'évaluation des méthodes d'ingénierie inverse. L'objectif principal est de coupler la théorie et l'expérience dans le domaine de l'inférence de réseau cellulaire et la construction de modèles quantitatifs en biologie des systèmes. Les questions fondamentales pour DREAM sont simples : Comment pouvons-nous évaluer la façon dont nous décrivons les réseaux d'interaction ? et comment évaluons-nous les résultats des expériences inédites de nos modèles ? Les réponses à ces questions ne sont pas si simples. En effet, des mesures non standardisées ont été appliqué et une variété d'algorithmes pour déduire la structure des réseaux biologiques et/ou prédire le résultat des perturbations des systèmes ont été utilisé. Cependant, DREAM a pour objectif de réaliser une comparaison "juste" des points forts et des points faibles des méthodes afin d'avoir un sens clair de la fiabilité des modèles.

DREAM organise un concours annuel d'ingénierie inverse [MCK+12, JMSR+10, SMC07, SPC09] qui comporte plusieurs challenges [2]. Généralement, un challenge DREAM vise à évaluer les réseaux de régulation inférés sur des données benchmark simulées et réelles, d'une manière double aveugle. La Figure **??** illustre la procédure d'évaluation en double aveugle (à l'aide de réseaux et de données synthétiques). Pour un challenge d'inférence de réseau, les organisateurs DREAM génèrent ou collectent plusieurs ensembles de données d'expression génique qui sont ensuite fournis aux équipes participantes. Pour chaque ensemble de données, l'objectif du challenge est de fournir une prédiction du réseau de régulation sous-jacent, sous la forme d'une liste de tous les potentiels liens (dirigées) de régulation classés selon leur indice du plus au moins confiant. En connaissant la base des réseaux réels, appelé la base *Gold*, plusieurs mesures d'évaluation du classement des liens de régulation correspondant à un réseau peuvent être estimées :

- AUROC : l'air de la courbe ROC
- AUPR : l'air de la courbe précision rappel PR
- AUROC p-value : la probabilité qu'une donnée AUROC est obtenue suite à un classement aléatoire de liens potentiels
- AUPR p-value : la probabilité qu'une donnée AUPR est obtenue suite à un classement aléatoire de liens potentiels

1. http ://wiki.c2b2.columbia.edu/dream/index.php/The_DREAM_Project
2. http ://wiki.c2b2.columbia.edu/dream/index.php/Challenges

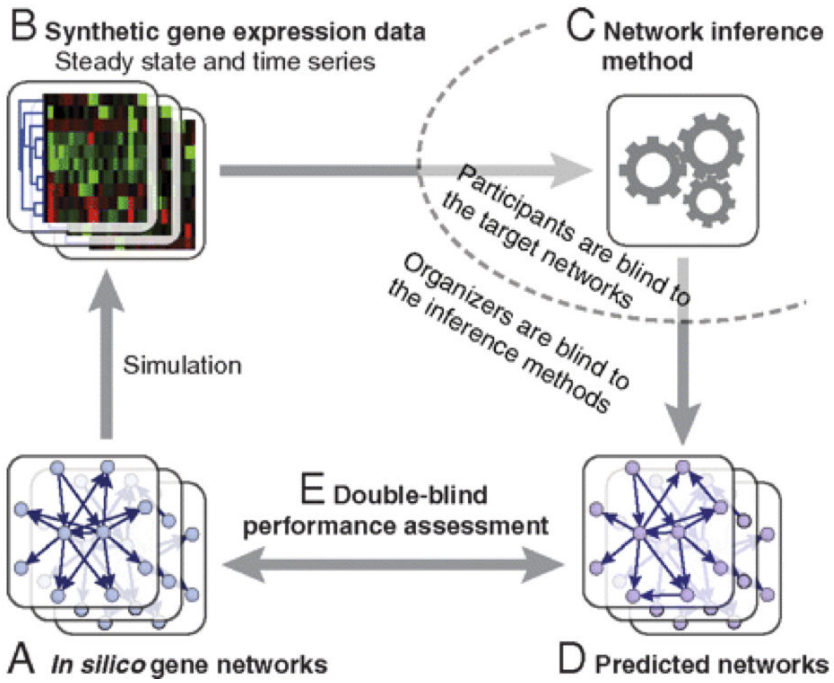


FIGURE B.3 – Évaluation de la performance des méthodes d'inférence de réseau en double aveugle. A. des réseaux de régulation génétique artificielle. B. Plusieurs données d'expression sont simulées à partir des réseaux artificiels et fournis aux participants de challenge. C. Les participants, ingorant des véritables réseaux, sont invités à les inférés à partir des données fournies. D. et E. Les organisateurs du challenge évaluent les prédiction de chaque participant, en ignorant l'algorithme d'inférence qui les les a générés. Figure extraite de [MCK+12]

Bibliographie

[AA03] E. Alm et A.P. Arkin : Biological networks. *Curr Opin Struct Biol*, 13 :193–202, 2003.

[ABpKS99] Mihael Ankerst, Markus M. Breunig, Hans peter Kriegel et Jörg Sander : Optics : Ordering points to identify the clustering structure. pages 49–60. ACM Press, 1999.

[AES10] Gökmen Altay et Frank Emmert-Streib : Inferring the conservative causal core of gene regulatory networks. *BMC Systems Biology*, 4(1) :132, 2010.

[AES11] Gökmen Altay et Frank Emmert-Streib : Structural influence of gene networks on their inference : analysis of c3net. *Biology direct*, 6(1) :31, 2011.

[AGW97] Y. Amit, D. Geman et K. Wilder : Joint induction of shape features and tree classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(11) :1300–1305, 1997.

[AHVdP+10] Thomas Abeel, Thibault Helleputte, Yves Van de Peer, Pierre Dupont et Yvan Saeys : Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3) : 392–398, 2010.

[AINH91] R. Jacobs A, M. Jordan I, S. Nowlan et G. Hinton : Adaptive mixtures of local experts. *Neural computation*, 3(1) :79–87, 1991.

[AIS93] R. Agrawal, T. Imielinski et A. Swami : Mining association rules between sets of items in large databases. *In Proceedings of the International Conference on Management of Data*, pages 207–216, 1993.

[AKMM98] T. Akutsu, S. Kuhara, O. Maruyama et S. Miyano : Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. *In Proc. Symposium on Discrete Algorithms (SODA)*, pages 695–702, 1998.

[AMK99] T. Akutsu, S. Miyano et S. Kuhara : Identification of genetic networks from a small number of gene expression patterns under the boolean network model. *In Pacific Symposium in Biocomputing*, pages 17–28, 1999.

[AS64] M. Abramowitz et I.A. Stegun : *Handbook of Mathematical Functions : With Formulars, Graphs, and Mathematical Tables*, volume 55. DoverPublications. com, 1964.

[BA97] L. Brelow et D. Aha : Simplifying decision trees : a survey. *The Knowledge Engineering Review*, 12 :1–40, 1997.

[Bac08] F. R. Bach : Bolasso : model consistent lasso estimation through the bootstrap. *In Proceedings of the 25th international conference on Machine learning*, pages 33–40. ACM, 2008.

[BdlFM02] P. Brazhnik, A. de la Fuente et P. Mendes : Gene networks : how to put the function in genomics. *Trends in Biotechnology*, 20 :467–472, 2002.

[BDP06] S. Kotsiantis B, L. Zaharakis D et P.E. Pintelas : Machine learning : a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3) :159–190, 2006.

[BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen et C. J. Stone : *Classification And Regression Trees*. Chapman and Hall, New York, 1984.

[BGH03] N. E. Buchler, U. Gerland et T. Hwa : On schemes of combinatorial transcription logic. *PNAS*, 100 :5136–5141, 2003.

[BH00] Y. Benjamini et Y. Hochberg : On the adaptive control of the false discovery rate in multiple testing with independent statistics. *Journal of Educational and Behavioral Statistics*, 25(1) :60–83, 2000.

[BH03] B. Bakker et T. Heskes : Clustering ensembles of neural network models. *Neural networks*, 16(2) :261–269, 2003.

[BHA09a] S. Bernard, L. Heutte et S. Adam : Influence of hyperparameters on random forest accuracy. *In Multiple Classifier Systems*, pages 171–180. Springer, 2009.

[BHA09b] S. Bernard, L. Heutte et S. Adam : On the selection of decision trees in random forests. *In Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 302–307. IEEE, 2009.

[BHBK05] R. Banfield, L. Hall, K. Bowyer et W. P. Kegelmeyer : Ensemble diversity measures and their application to thinning. *Information Fusion*, 6(1) :49–62, 2005.

[BK99] E. Bauer et R. Kohavi : An empirical comparison of voting classification algorithms : Bagging, boosting, and variants. *Machine learning*, 36(1-2) :105–139, 1999.

[BK00] A.J. Butte et I.S. Kohane : Mutual information relevance networks : functional genomic clustering using pairwise entropy measurements. *Pac Symp Biocomput*, pages 418–29, 2000.

[BMS+05] K. Basso, A.A. Margolin, G. Stolovitzky, U. Klein, R. Dalla-Favera et A. Califano : Reverse engineering of regulatory networks in human B cells. *Nat Genet*, 2005.

[BN00] D. Bahler et L. Navarro : Methods for combining heterogeneous sets of classifiers. *In 17th Natl. Conf. on Artificial Intelligence (AAAI), Workshop on New Research Problems for Machine Learning*, 2000.

[BN06] C. M. Bishop et N. M. Nasrabadi : *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[BND04] U.M. Braga-Neto et E.R. Dougherty : Is cross-validation valid for small-sample microarray classification ? *Bioinformatics*, 20(3) :374–380, 2004.

[BO04] A.L. Barabasi et Z.N. Oltvai : Network biology : understanding the cell's functional organization. *Nature Reviews Genetics*, 5 :101–113, 2004.

[Bra97] A.P. Bradley : The use of the area under the ROC curve in the evaluation of machine learning algorithms. *In Pattern Recognition*, 1997.

[Bre96a] L. Breiman : Bagging predictors. *Machine Learning*, 24(2) :123–140, 1996.

[Bre96b] L. Breiman : Stacked regressions. *Machine learning*, 24(1) :49–64, 1996.

[Bre98] L. Breiman : Arcing classifier. *The annals of statistics*, 26(3) :801–849, 1998.

[Bre00] L. Breiman : Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40(3) :229–242, 2000.

[Bre01] L. Breiman : Random forests. *Machine Learning*, 45(1) :5–32, 2001.

[BS09] Anne-Laure Boulesteix et Martin Slawski : Stability and aggregation of ranked gene lists. *Briefings in bioinformatics*, 10(5) :556–568, 2009.

[Büh12] Peter Bühlmann : Bagging, boosting and ensemble methods. *In Handbook of Computational Statistics*, pages 985–1022. Springer, 2012.

[BWHY05] G. Brown, J. Wyatt, R. Harris et X. Yao : Diversity creation methods : a survey and categorisation. *Information Fusion*, 6(1) :5–20, 2005.

[C+60] Jacob Cohen *et al.* : A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1) :37–46, 1960.

[Car98] Michael Carey : The enhanceosome and transcriptional synergy. *Cell*, 92(1) :5–8, 1998.

[CEN+] Ines Chebil, Mohamed Elati, Rémy Nicolle, Christophe Rodrigues et Céline Rouveirol : Licorn* : construction de réseaux de régulation chez l'homme.

[CERS13] I. Chebil, M. Elati, C. Rouveirol et G. Santini : Setnet : Ensemble method techniques for learning regulatory networks. *In Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 1, pages 34–39, Dec 2013.

[Che96] K.J. Cherkauer : Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks. *In Working notes of the AAAI workshop on integrating multiple learned models*, pages 15–21. Citeseer, 1996.

[Chi02] D. M. Chickering : Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3 :507–554, 2002.

[CHM04] D. M. Chickering, D. Heckerman et C. Meek : Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5 :1287–1330, 2004.

[CM02] A. Cornuéjols et L. Miclet : *Apprentissage artificiel : Concepts et algorithmes*. Eyrolles, 2002.

[CMS+06] Jason S Carroll, Clifford A Meyer, Jun Song, Wei Li, Timothy R Geistlinger, Jérôme Eeckhoute, Alexander S Brodsky, Erika Krasnickas Keeton, Kirsten C Fertuck, Giles F Hall *et al.* : Genome-wide analysis of estrogen receptor binding sites. *Nature genetics*, 38(11) :1289–1297, 2006.

[CNCK04] R. Caruana, A. Niculescu, G. Crew et A. Ksikes : Ensemble selection from libraries of models. *In Proceedings of the twenty-first international conference on Machine learning*, page 18. ACM, 2004.

[CNS+13] I. Chebil, R. Nicolle, G. Santini, C. Rouveirol et M. Elati : Hybrid method inference for the construction of cooperative regulatory network in human. *In Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on*, pages 121–126, Dec 2013.

[CNS+14] I Chebil, R Nicolle, G Santini, C Rouveirol et M Elati : Hybrid method inference for the construction of cooperative regulatory network in human. *IEEE transactions on nanobioscience*, 2014.

[Cor09] H. Cordell : Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics*, 10(6) :392–404, 2009.

[CVZ06] G. Coelho et J. Fernando Von Zuben : The influence of the pool of candidates on the performance of selection and combination techniques in ensembles. *In Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 5132–5139. IEEE, 2006.

[CWC06] Y.H. Chang, Y.C. Wang et B.S. Chen : Identification of transcription factor cooperativity via stochastic system model. *Bioinformatics*, 22 :2276–2282, 2006.

[DB95] T. Dietterich. et G. Bakiri : Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101*, 1995.

[DCC+12] David J DeGraff, Peter E Clark, Justin M Cates, Hironobu Yamashita, Victoria L Robinson, Xiuping Yu, Mark E Smolkin, Sam S Chang, Michael S Cookson, Mary K Herrick, Shahrokh F Shariat, Gary D Steinberg, Henry F Frierson, Xue-Ru Wu, Dan Theodorescu et Robert J Matusik : Loss of the Urothelial Differentiation Marker FOXA1 Is Associated with High Grade, Late Stage Bladder Cancer and Increased Tumor Proliferation. *PLoS ONE*, 7 :e36669, 2012.

[Det04] M. Dettling : Bagboosting for tumor classification with gene expression data. *Bioinformatics*, 20(18) :3583–3593, 2004.

[DIB97] J.L. DeRisi, V.R. Iyer et B.O. Brown : Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278 :680–686, 1997.

[Die97] Thomas G Dietterich : Machine learning research : four current direction. *Artificial Intelligence Magzine*, 4 :97–136, 1997.

[Die00a] T. Dietterich : Ensemble methods in machine learning. *Multiple Classifier Systems*, 1857 :1–15, 2000.

[Die00b] T.G. Dietterich : An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, boosting, and randomization. *Machine learning*, 40(2) :139–157, 2000.

[DK95] T. Dietterich et E. Kong : Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. *ML-95*, 255, 1995.

[DK02] F. Dardel et F. K»pÀs : *Bioinformatique, g»nomique et post g»nomique*. Editions de l'Ecole Polytechnique, 2002.

[dlFBHM04] A. de la Fuente, N. Bing, I. Hoeschele et P. Mendes : Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20 :3565–3574, 2004.

[DLS00] P. D'haeseleer, S. Liang et R. Somogyi : Genetic network inference : From co-expression clustering to reverse engineering. *Bioinformatics*, 16 :707–726, 2000.

[DMSES12] Ricardo De Matos Simoes et Frank Emmert-Streib : Bagging statistical network inference from large-scale gene expression data. *PLoS One*, 7(3) :e33624, 2012.

[DP05] Chris Ding et Hanchuan Peng : Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02) :185–205, 2005.

[Dra03] S. Draghici : *Data Analysis Tools for DNA Microarrays*. Chapman & Hall/CRC, 2003.

[Dru97] H. Drucker : Improving regressors using boosting techniques. *In ICML*, volume 97, pages 107–115, 1997.

[DT00] R.PW. Duin et D. MJ Tax : Experiments with classifier combining rules. *In Multiple Classifier Systems*, pages 16–29. Springer, 2000.

[DUDA06] Ramón Díaz-Uriarte et Sara Alvarez De Andres : Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1) :3, 2006.

[EHJT04] B. Efron, Trevor H., L. Johnstone et R. Tibshirani : Least angle regression. *The Annals of statistics*, 32(2) :407–499, 2004.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander et Xiaowei Xu : A density-based algorithm for discovering clusters in large spatial databases with noise. *In KDD*, volume 96, pages 226–231, 1996.

[ENBF+07] M. Elati, P. Neuvial, M. Bolotin-Fukuhara, E. Barillot, F. Radvanyi et C. Rouveirol : Licorn : learning cooperative regulation networks from gene expression data. *Bioinformatics*, 23 :2407–2414, 2007.

[ET94] B. Efron et R. Tibshirani : An introduction to the bootstrap (chapman & hall/crc monographs on statistics & applied probability). 1994.

[FB03] X.Z. Fern et C.E. Brodley : Random projection for high dimensional data clustering : A cluster ensemble approach. *In ICML*, volume 3, pages 186–193, 2003.

[FHT+07] J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins et T. S. Gardner : Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol*, 5 :e8, 2007.

[FKT00] A. W. C. Fu, R. Kwong et J. Tang : Mining n-most interesting itemsets. *In ISMIS*, pages 59–67, 2000.

[FLNP00] N. Friedman, M. Linial, I. Nachman et D. Pe'er : Using bayesian network to analyze expression data. *Computational Biology*, 7 :601–620, 2000.

[FR05] G. Fumera et F. Roli : A theoretical and experimental analysis of linear combiners for multiple classifier systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6) :942–956, 2005.

[Fri01] J.H. Friedman : Greedy function approximation : a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[Fri04] N. Friedman : Inferring cellular networks using probabilistic graphical models. *Science*, 303 :799–805, 2004.

[FS95] Y. Freund et R. Schapire : A desicion-theoretic generalization of on-line learning and an application to boosting. *In Computational learning theory*, pages 23–37. Springer, 1995.

[FS+96] Y. Freund, R. Schapire *et al.* : Experiments with a new boosting algorithm. *In ICML*, volume 96, pages 148–156, 1996.

[FSF+12] A Franceschini, D Szklarczyk, S Frankild, M Kuhn, M Simonovic, A Roth, J Lin, P Minguez, P Bork, C von Mering et L J Jensen : STRING v9.1 : protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41 :D808–D815, 2012.

[FW00] James W Fickett et Wyeth W Wasserman : Discovery and modeling of transcriptional regulatory regions. *Current Opinion in Biotechnology*, 11(1) :19–24, 2000.

[GB00] J. Gama et P. Brazdil : Cascade generalization. *Machine Learning*, 41(3) :315–343, 2000.

[GDDP09] D. Gacquer, V. Delcroix, F. Delmotte et S. Piechowiak : On the effectiveness of diversity when training multiple classifier systems. *In Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 493–504. Springer, 2009.

[GEW06] P. Geurts, D. Ernst et L. Wehenkel : Extremely randomized trees. *Machine learning*, 63(1) :3–42, 2006.

[GF05] T. S. Gardner et J. J. Faith : Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2 :65–88, 2005.

[GH10] M. Gustafsson et M. Hörnquist : Gene expression prediction by soft integration and the elastic net a best performance of the dream3 gene expression challenge. *PLoS ONE*, 5(2) :e9134, 02 2010.

[GHL05] M. Gustafsson, M. Hörnquist et A. Lombardi : Constructing and analyzing a large-scale gene-to-gene regulatory network-lasso-constrained inference and biological validation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(3) :254–261, 2005.

[GMMS08] D. Hernández-Lobato G. Martínez-Muñoz, A. Sánchez-Martínez et A. Suárez : Class-switching neural network ensembles. *Neurocomputing*, 71(13) :2521–2528, 2008.

[GR01] G. Giacinto et F. Roli : An approach to the automatic design of multiple classifier systems. *Pattern recognition letters*, 22(1) :25–33, 2001.

[GRF00] G. Giacinto, F. Roli et G. Fumera : Design of effective multiple classifier systems by clustering of classifiers. *In Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 2, pages 160–163. IEEE, 2000.

[GS02] Larry D. Greller et Roland Somogyi : Reverse engineers map the molecular switching yards. *Trends in Biotechnology*, 20(11) :445 – 447, 2002.

[GTDdB07] P. Geurts, N. Touleimat, M. Dutreix et F. d'Alch» Buc : Inferring biological networks with output kernel trees. *BMC Bioinformatics*, 8, 2007.

[GW95a] M. Goemans et D. Williamson : Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6) :1115–1145, 1995.

[GW95b] Michel X. Goemans et David P. Williamson : Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6) :1115–1145, novembre 1995.

[GWdAB06] P. Geurts, L. Wehenkel et F. d Alché-Buc : Ok3 : Méthode d'arbres à sortie noyau pour la prédiction de sorties structurées et l'apprentissage de noyau. *In Proc. of CAP*, page 16, 2006.

[Has97] C. Hashem : Optimal linear combinations of neural networks. *Neural networks*, 10(4) :599–614, 1997.

[HCG+00] Marc S Halfon, Ana Carmena, Stephen Gisselbrecht, Charles M Sackerson, Fernando Jiménez, Mary K Baylies et Alan M Michelson : Ras pathway specificity is determined by the integration of multiple signal-activated and tissue-restricted transcription factors. *Cell*, 103(1) :63–74, 2000.

[HGL+04] C.T. Harbison, D.B. Gordon, T.I. Lee, N.J. Rinaldi, K.D. Macisaac, T.W. Danford, N.M. Hannett, J.B. Tagne, D.B. Reynolds, J. Yoo, E.G. Jennings, J. Zeitlinger, D.K. Pokholok, M. Kellis, P.A. Rolfe, K.T. Takusagawa, E.S. Lander, D.K. Gifford, E. Fraenkel et R.A. Young : Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431 :99–104, 2004.

[HGV11] Anne-Claire Haury, Pierre Gestraud et Jean-Philippe Vert : The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PloS one*, 6(12) :e28210, 2011.

[HK08] M. Hilario et A. Kalousis : Approaches to dimensionality reduction in proteomic biomarker studies. *Briefings in Bioinformatics*, 9(2) :102–118, 2008.

[HLHLRTV06] D. Hernández-Lobato, J. Hernández-Lobato, R. Ruiz-Torrubiano et A. Valle : Pruning adaptive boosting ensembles by means of a genetic algorithm. *In Intelligent Data Engineering and Automated Learning–IDEAL 2006*, pages 322–329. Springer, 2006.

[HMVLV12] A. C. Haury, F. Mordelet, P. Vera-Licona et J. Vert : Tigress : trustful inference of gene regulation using stability selection. *BMC systems biology*, 6(1) :145, 2012.

[Ho98] Tin Kam Ho : The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8) :832–844, 1998.

[HS90] L. K. Hansen et P. Salamon : Neural network ensembles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(10) :993–1001, 1990.

[HS96] W.S. Hlavacek et M.A. Savageau : Rules for coupled expression of regulator and effector genes in inducible circuits. *J Mol Biol*, 255 :121–139, 1996.

[HS02] C.E. Horak et M. Snyder : Chip-chip : a genomic approach for identifying transcription factor binding sites. *Methods Enzymol*, 350 :469–483, 2002.

[HTF09] T. Hastie, R. Tibshirani et J. H. Friedman : *The elements of statistical learning : data mining, inference, and prediction, 2nd ed.* Springer-Verlag, 2009.

[HTFF05] T. Hastie, R. Tibshirani, J. Friedman et J. Franklin : The elements of statistical learning : data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2) :83–85, 2005.

[HTIWG10] Vân A. Huynh-Thu, Alexandre Irrthum, Louis Wehenkel et Pierre Geurts : Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. *PLoS ONE*, 5(9) :e12776+, septembre 2010.

[HTLF01] T. Hastie, R. Tibshirani et J Jerome L. Friedman : *The elements of statistical learning*, volume 1. Springer New York, 2001.

[HYZ02] Q. Han, Y. Ye et J. Zhang : An improved rounding method and semidefinite programming relaxation for graph partition. *Mathematical Programming*, 92(3) :509–535, 2002.

[ICD01] L. Kuncheva I, J. Bezdek C et R. Duin : Decision templates for multiple classifier fusion : an experimental comparison. *Pattern Recognition*, 34(2) : 299–314, 2001.

[IJ94] M. Jordan I et R. A. Jacobs : Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2) :181–214, 1994.

[IWAD03] L. Kuncheva I, C. Whitaker, C. Shipp A et R. Duin : Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis & Applications*, 6(1) :22–31, 2003.

[JM61] F. Jacob et J. Monod : Genetic regulatory mechanisms in the synthesis of proteins. *Molecular Biology*, 3(3) :318–356, 1961.

[JMSR+10] R. Prill J, D. Marbach, J. Saez-Rodriguez, P. Sorger K, L. Alexopoulos, X. Xue, N. Clarke, G. Altan-Bonnet et G. Stolovitzky : Towards a rigorous assessment of systems biology models : the dream3 challenges. *PloS one*, 5(2) :e9202, 2010.

[Kau69a] S. A. KAUFFMAN : Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22 :437–467, 1969.

[Kau69b] S. A. KAUFFMAN : Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22 :437–467, 1969.

[KDM00] A. Jain K, R. DUIN et J. MAO : Statistical pattern recognition : A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1) :4–37, 2000.

[KHDM98] J. KITTLER, M. HATEF, R. DUIN et J. MATAS : On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3) : 226–239, 1998.

[Kit98] J. KITTLER : Combining classifiers : A theoretical framework. *Pattern analysis and Applications*, 1(1) :18–27, 1998.

[KJS94] T. Ho KAM, J. Hull J. et S.N. SRIHARI : Decision combination in multiple classifier systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(1) :66–75, 1994.

[KLL+11] Say Li KONG, Guoliang LI, Siang Lin LOH, Wing-Kin SUNG et Edison T LIU : Cellular reprogramming by the conjoint action of ER&alpha ;, FOXA1, and GATA3 to a ligand-inducible growth state. *Molecular Systems Biology*, 7 :1–14, 2011.

[Koh95] KOHONEN : *Self-Organizing Maps*. Springer Verlag, Berlin, 1995.

[KP05] S. KOTSIANTIS et P. PINTELAS : Selective averaging of regression models. *Annals of Mathematics, Computing & Teleinformatics*, 1(3) :65–74, 2005.

[KPJ+03] H. KIM, S. PANG, H. JE, D. KIM et S. Yang BANG : Constructing support vector machine ensemble. *Pattern recognition*, 36(12) :2757–2767, 2003.

[KRK+95] Olga V KEL, Aida G ROMASCHENKO, Alexander E KEL, Edgar WINGENDER et Nikolay A KOLCHANOV : A compilation of composite regulatory elements affecting gene transcription in vertebrates. *Nucleic acids research*, 23(20) :4097–4103, 1995.

[Kun07] L. L. KUNCHEVA : Combining pattern classifiers : Methods and algorithms (kuncheva, li ; 2004)[book review]. *Neural Networks, IEEE Transactions on*, 18(3) :964–964, 2007.

[La96] D. J. LOCKHART et AL. : Expression monitoring by hybridization to high–density oligonucleotide arrays. *Nature Biotechnology*, 14(13) :1675–1680, 1996.

[LFS98] S. LIANG, S. FUHRMAN et R. SOMOGYI : Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *In Pacific Symposium in Biocomputing*, pages 18–29, 1998.

[LGLM+06] Guillaume LAMIRAULT, Nathalie GABORIT, Nolwenn LE MEUR, Catherine CHEVALIER, Gilles LANDE, Sophie DEMOLOMBE, Denis ESCANDE,

Stanley Nattel, Jean J Léger et Marja Steenman : Gene expression profile associated with chronic atrial fibrillation and underlying valvular heart disease in man. *Journal of molecular and cellular cardiology*, 40(1) :173–184, 2006.

[LJFJ06] L.Elnitski, V.X Jin, P.J Farnham et S.J.M Jones : Locating mammalian transcription factor binding sites : A survey of computational and experimental techniques. *Genome Res*, 2006.

[LLPS05] J.W. Lee, J. B. Lee, M. Park et S. Song : An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis*, 48(4) :869–885, 2005.

[LO01] A. Lazarevic et Z. Obradovic : Effective pruning of neural network classifier ensembles. *In Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on*, volume 2, pages 796–801. IEEE, 2001.

[LRA+10] Celine Lefebvre, Presha Rajbhandari, Mariano J Alvarez, Pradeep Bandaru, Wei Keat Lim, Mai Sato, Kai Wang, Pavel Sumazin, Manjunath Kustagi, Brygida C Bisikirska, Katia Basso, Pedro Beltrao, Nevan Krogan, Jean Gautier, Riccardo Dalla-Favera et Andrea Califano : A human B-cell interactome identifies MYB and FOXM1 as master regulators of proliferation in germinal centers. *Molecular Systems Biology*, 6 :1–10, 2010.

[LRR+02] T.I. Lee, N. J. Rinaldi, F. R., D.T Odom, Z. Bar-Joseph, G.K. Gerber, N.M. Hannett, C.T. Harbison, C.M. Thompson, I. Simon, J. Zeitlinger, E.G Jennings, H.L. Murray, D.B. Gordon, B. Ren, J. J. Wyrick, J.B Tagne, T.L. Volkert, E. Fraenkel, D.K. Gifford et R.A Young : Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, 298 :799–804, 2002.

[LS97] L. Lam et S. Suen : Application of majority voting to pattern recognition : an analysis of its behavior and performance. *Systems, Man and Cybernetics, Part A : Systems and Humans, IEEE Transactions on*, 27(5) :553–568, 1997.

[LSYH03] H. Lähdesmäki, I. Shmulevich et O. Yli-Harja : On Learning Gene Regulatory Networks Under the Boolean Network Model. *Machine Learning*, 52 :147–167, 2003.

[LT03] Michael Levine et Robert Tjian : Transcription regulation and animal diversity. *Nature*, 424(6945) :147–151, 2003.

[LY99] Y. Liu et X. Yao : Ensemble learning via negative correlation. *Neural Networks*, 12(10) :1399–1404, 1999.

[LZK13] Dongye Liu, Zhe Zhang et Chui-ze Kong : High foxm1 expression was associated with bladder carcinogenesis. *Tumor Biology*, 34 :1131–1138, 2013.

[MB10] N. Meinshausen et P. Bühlmann : Stability selection. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 72(4) :417–473, 2010.

[McC98] J. E. ?G. McCarthy : Posttranscriptional Control of Gene Expression in Yeast. *Microbiol. Mol. Biol. Rev.*, 62 :1492–1553, 1998.

[MCK+12] D. Marbach, J. C. Costello, R. Küffner, N. Vega, R. J. Prill, D. Camacho, K. R. Allison R, M. Kellis, J. Collins, G. Stolovitzky *et al.* : Wisdom of crowds for robust gene network inference. *Nature methods*, 2012.

[MD97] D. Margineantu et T. Dietterich : Pruning adaptive boosting. *In ICML*, volume 97, pages 211–218. Citeseer, 1997.

[MdJFSJS06] M. J. Moreira, de J. F. SOUSA, A. M. Jorge et C. Soares : An ensemble regression approach for bus trip time prediction. *In Proceedings of the EWGT2006 joint conferences*, 2006.

[MFG+03] V. Matys, E. Fricke, R. Geffers, E. Gossling, M. Haubrock, R. Hehl, K. Hornischer, D. Karas, A. E. Kel, O. V. Kel-Margoulis et D. U. Kloos : Transfac : transcriptional regulation, from patterns to profiles. *Nucleic Acids Research*, 31 :374–378, 2003.

[MG09] M Hörnquist M Gustafsson : *Integrating various data sources for improved quality in reverse of gene regulatory networks.* In : Das S, Caragea D, Hsu WH, Welch SM, 2009.

[MGT09] J. Lundström J. Björkegren M. Gustafsson, M. Hörnquist et J. Tegnér : Reverse engineering of gene networks with lasso and nonlinear basis functions. *Annals of the New York Academy of Sciences*, 1158(1) :265–275, 2009.

[MHADI06] D. Colak M. H. Asyali, O. Demirkaya et M. Inan : Gene expression profile classification : a review. *Current Bioinformatics*, 1(1) :55–73, 2006.

[Mic83] R. Michalski : A theory and methodology for inductive learning. *In Machine Learning : An artificial intelligence approach.* 1983. 83-134.

[Mit77] T. M. Mitchell : Version spaces : a candidate elimination approach to rule learning. *In the 5th International Joint Conference on Artificial Intelligence*, pages 305–310, 1977.

[Mit82] T. M. Mitchell : Generalization as search. *Artificial Intelligence*, 18 :203–226, 1982.

[MIT03] 10 emerging technologies that will change the world. *MIT Technology Review*, 2003.

[MKLB07] P. E. Meyer, K. Kontos, F. Lafitte et G. Bontempi : Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J. Bioinformatics and Systems Biology*, 2007.

[MKMF+06] V Matys, O V Kel-Margoulis, E Fricke, I Liebich, S Land, A Barre-Dirrie, I Reuter, D Chekmenev, M Krull, K Hornischer, N Voss, P Stegmaier, B Lewicki-Potapov, H Saxel, A E Kel et E Wingender : TRANSFAC and its module TRANSCompel : transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, 34 :D108–10, 2006.

[MKW+04] M. Middendorf, A. Kundaje, C. Wiggins, Y. Freund et C. Leslie : Predicting genetic regulatory response using classification. *Bioinformatics*, 20 :232–240, 2004.

[MLB08] P. Emmanuel Meyer, F. Lafitte et G. Bontempi : *Minet* : A R/Bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics*, 2008.

[MMHLS09] G. Martinez-Muoz, D. Hernández-Lobato et A. Suárez : An analysis of ensemble pruning techniques based on ordered aggregation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2) : 245–259, 2009.

[MMS05] G. Martínez-Muñoz et A. Suárez : Switching class labels to generate classification ensembles. *Pattern Recognition*, 38(10) :1483–1494, 2005.

[MMS06] G. Martínez-Muñoz et A. Suárez : Pruning in ordered bagging ensembles. *In Proceedings of the 23rd international conference on Machine learning*, pages 609–616. ACM, 2006.

[MNB+06] A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Favera et A. Califano : Aracne : An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7 :S7, 2006.

[MO11] R. Maclin et D. Opitz : Popular ensemble methods : An empirical study. *arXiv preprint arXiv :1106.0257*, 2011.

[MR03] R. Meir et G. Rätsch : An introduction to boosting and leveraging. *In Advanced lectures on machine learning*, pages 118–183. Springer, 2003.

[MSM+10] D. Marbach, T. Schaffter, C. Mattiussi, D. Floreano et G. Stolovitzky : Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences*, 107(14) :6286–6291, 2010.

[MSMF09] D. Marbach, T. Schaffter, C. Mattiussi et D. Floreano : Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology*, 16(2) :229–239, 2009.

[MT07] J. Meynet et J.P. Thiran : Information theoretic combination of classifiers with application to adaboost. *In Multiple Classifier Systems*, pages 171–179. Springer, 2007.

[MV08] Fantine Mordelet et Jean-Philippe Vert : Sirene : supervised inference of regulatory networks. *Bioinformatics*, 24(16) :i76–i82, 2008.

[NKS05] N. Nagamine, Y. Kawada et Y. Sakakibara : Identifying cooperative transcriptional regulations using protein-protein interactions. *Nucl. Acids Res.*, 33 :4828–4837, 2005.

[NPW99] D. Politis N, J. Romano P et M. Wolf : *Subsampling for nonstationary time series*. Springer, 1999.

[Orl00] Valerio Orlando : Mapping chromosomal proteins¡ i¿ in vivo¡/i¿ by formaldehyde-crosslinked-chromatin immunoprecipitation. *Trends in biochemical sciences*, 25(3) :99–104, 2000.

[PC92] M. Perrone P et L.N Cooper : When networks disagree : Ensemble methods for hybrid neural networks. Rapport technique, DTIC Document, 1992.

[Pea78] J. Pearl : On the connection between the complexity and credibility of inferred models. *International Journal of General Systems*, 4 :255–264, 1978.

[PRT02] D. Pe'er, A. Regev et A. Tanay : Minreg : inferring an active regulator set. *Bioinformatics*, 18 :258–267, 2002.

[PTKV06] L. Partalas, G. Tsoumakas, L. Katakis et L. Vlahavas : Ensemble pruning using reinforcement learning. *In Advances in Artificial Intelligence*, pages 301–310. Springer, 2006.

[PTR06] D. Pe'er, A. Tanay et A. Regev : Minreg : A scalable algorithm for learning parsimonious regulatory networks in yeast and mammals. *Journal of Machine Learning Research*, 7 :167–189, 2006.

[PY96] D. Partridge et W. Yates : Engineering multiversion neural-net systems. *Neural Computation*, 8(4) :869–893, 1996.

[QLYs03] J. Qian, J. Lin, H. Yu et M. stein : Prediction of regulatory networks : genome-wide identification of transcription factor targets from gene expression data. *Bioinformatics*, 19 :17–26, 2003.

[RDS10] Kathleen Marchal Riet De Smet : Advantages and limitations of current network inference methods. *Nature Reviews Microbiology*, (10) :717–729, 2010.

[RGV01] F. Roli, G. Giacinto et G. Vernazza : *Methods for designing multiple classifier systems*. Springer, 2001.

[RIA06] J. J. Rodriguez, L. Kuncheva I et C. J. Alonso : Rotation forest : A new classifier ensemble method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10) :1619–1630, 2006.

[RP11] P Raychaudhuri et H J Park : FoxM1 : A Master Regulator of Tumor Metastasis. *Cancer Research*, 71(13) :4329–4333, 2011.

[RPAT04] N. Rooney, D. Patterson, S. Anand et A. Tsymbal : Dynamic integration of regression models. *In Multiple Classifier Systems*, pages 164–173. Springer, 2004.

[RRW+00] Bing Ren, François Robert, John J Wyrick, Oscar Aparicio, Ezra G Jennings, Itamar Simon, Julia Zeitlinger, Jörg Schreiber, Nancy Hannett, Elenita Kanin *et al.* : Genome-wide location and function of dna binding proteins. *Science*, 290(5500) :2306–2309, 2000.

[RSB03] B. Dolenko R.L Somorjai et R. Baumgartner : Class prediction and discovery using gene microarray and proteomics mass spectroscopy data : curses, caveats, cautions. *Bioinformatics*, 19(12) :1484–1491, 2003.

[Sa06] N. Stransky et al. : Regional copy number-independent deregulation of transcription in cancer. *Nature Genetics*, 38 :1386–1396, 2006.

[SA12] J. Slawek et T. Arodź : ADANET : inferring gene regulatory networks using ensemble classifiers. *In Proceedings BCB '12*, pages 434–441, 2012.

[SAB+77] F. Sanger, G.M. Air, B.G. Barrell, N.L. Brown, A.R. Coulson, C.A. Fiddes, C.A. Hutchison, P.M. Slocombe et M. Smith : Nucliotide sequence of bacteriophage phi x174 dna. *Nature*, 265 :687–95, 1977.

[SAVdP08] Yvan Saeys, Thomas Abeel et Yves Van de Peer : Robust feature selection using ensemble feature selection techniques. *In Machine Learning and Knowledge Discovery in Databases*, pages 313–325. Springer, 2008.

[SB07] T. Schlitt et A. Brazma : Current approaches to gene regulatory network modelling. *BMC bioinformatics*, 8(Suppl 6) :S9, 2007.

[SBE99] B. Schölkopf, C. Burges et A. Smola (Eds) : *Advances in kernel methods. Support vector learning*. MIT Press, 1999.

[Sch94] C. Schaffer : A conservation law for generalization performance. *In ICML*, volume 94, pages 259–265, 1994.

[Sch97] R.E. Schapire : Using output codes to boost multiclass learning problems. *In ICML*, volume 97, pages 313–321, 1997.

[SD02] M. Skurichina et R. Duin : Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2) :121–135, 2002.

[SD05] M. Skurichina et R. PW. Duin : Combining feature subsets in feature selection. *In Multiple classifier systems*, pages 165–175. Springer, 2005.

[SFBL98] R. Schapire, Y. Freund, P. Bartlett et L.S Lee : Boosting the margin : A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5) :1651–1686, 1998.

[SGS+00] P. Spirtes, C. Glymour, R. Scheines, S. Kauffman, V. Aimale et F. Wimberly : Constructing Bayesian network models of gene expression networks from microarray data. *In Proc. of the Atlantic Symposium on Computational Biology, Genome Information Systems and Technology*, 2000.

[SHA96] A. SHARKEY : On combining artificial neural nets. *Connection Science*, 8(3-4) :299–314, 1996.

[SJH02] V. A. Smith, E. D. Jarvis et A. J. Hartemink : Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics*, 18 :216S–224, 2002.

[SK99] R. Somogyi et H. Kitano : Gene expression and genetic networks. *In Pacific Symposium in Biocomputing*, pages 3–4, 1999.

[SKB+03] Lev A Soinov, Maria A Krestyaninova, Alvis Brazma *et al.* : Towards reconstruction of gene networks from expression data by supervised learning. *Genome biology*, 4(1) :R6, 2003.

[SL91] S. Rasoul Safavian et D. Landgrebe : A survey of decision tree classifier methodology. *IEEE Transactions on SMC*, 3, 1991.

[SMC07] Gustavo Stolovitzky, DON Monroe et Andrea Califano : Dialogue on reverse-engineering assessment and methods. *Annals of the New York Academy of Sciences*, 1115(1) :1–22, 2007.

[SPC09] G Stolovitzky, RJ Prill et A Califano : Lessons from the dream2 challenges. *Annals of the New York Academy of Sciences*, 1158, 03 2009.

[SS97] A. Sharkey et N.E Sharkey : Combining diverse neural nets. *Knowledge Engineering Review*, 12(3) :231–247, 1997.

[SS02] B. Schölkopf et A. Smola : *Learning with kernels. Support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.

[SSDB95] M. Schena, D. Shalon, R. Davis et P. Brown : Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 70 :467–470, 1995.

[SSR+03] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller et N. Friedman : Module networks : identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34 :166–176, 2003.

[SSZ+98] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer VRand K. Anders, M.B. Eisen, P.O. Brown, D. Botstein et B. Futcher : Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9 :3273–97, 1998.

[TAV05] G. Tsoumakas, L. Angelis et I. Vlahavas : Selective fusion of heterogeneous classifiers. *Intelligent Data Analysis*, 9(6) :511–525, 2005.

[TD03] L. Todorovski et S. Džeroski : Combining classifiers with meta decision trees. *Machine learning*, 50(3) :223–249, 2003.

[TG96] Kagan Tumer et Joydeep Ghosh : Error correlation and error reduction in ensemble classifiers. *Connection science*, 8(3-4) :385–404, 1996.

[Tib94] Robert Tibshirani : Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58 :267–288, 1994.

[TKV04] G. Tsoumakas, L. Katakis et L. Vlahavas : Effective voting of heterogeneous classifiers. *In Machine Learning : ECML 2004*, pages 465–476. Springer, 2004.

[TX00] C. Tamon et J. Xiang : On the boosting pruning problem. *In Machine Learning : ECML 2000*, pages 404–412. Springer, 2000.

[UN96] N. Ueda et R. Nakano : Generalization error of ensemble estimators. *In Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 90–95. IEEE, 1996.

[Vap98] V. Vapnik : *Statistical learning theory*. Wiley-InterScience, 1998.

[Vap00] V. Vapnik : *The nature of statistical learning theory*. springer, 2000.

[VDFV03] Roel Van Driel, Paul F Fransz et Pernette J Verschure : The eukaryotic genome : a system regulated at different hierarchical levels. *Journal of Cell Science*, 116(20) :4067–4075, 2003.

[VM02] G. Valentini et F. Masulli : Ensembles of learning machines. *In Neural Nets*, pages 3–20. Springer, 2002.

[VVA+11] Matthieu Vignes, Jimmy Vandel, David Allouche, Nidal Ramadan-Alban, Christine Cierco-Ayrolles, Thomas Schiex, Brigitte Mangin et Simon de Givry : Gene regulatory network reconstruction using bayesian networks, the dantzig selector, the lasso and their meta-analysis. *PloS one*, 6(12) :e29165, 2011.

[WB98] C. Williams et D. Barber : Bayesian classification with gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12) :1342–1351, 1998.

[WBD+01] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. Olson A, J. Marks R et J. R. Nevins : Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences*, 98(20) :11462–11467, 2001.

[Web00] G. I. Webb : Multiboosting : A technique for combining boosting and wagging. *Machine learning*, 40(2) :159–196, 2000.

[Wei05] S. Weisberg : *Applied linear regression*, volume 528. Wiley. com, 2005.

[WGH06] A. V Werhli, M. Grzegorczyk et D. Husmeier : Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics*, 22 :2523–2531, 2006.

[WHRG03] F.C Wimberly, T. Heiman, J. Ramsey et C. Glymour : Experiments on the accuracy of algorithms for inferring the structure of genetic regulatory networks from microarray expression levels. *In Proc. IJCAI 2003 Bioinformatics Workshop*, 2003.

[Wol96] D. H. Wolpert : The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7) :1341–1390, 1996.

[WYH03] F. W. Wangand, P.S. Yu et J. Han : Mining concept-drifting data streams using ensemble classifiers. *In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235. ACM, 2003.

[XKS92] L. Xu, A. Krzyzak et C. Y. Suen : Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(3) :418–435, 1992.

[YSL07] I. Inza Y. Saeys et P. Larrañaga : A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19) :2507–2517, 2007.

[ZBS06] Y. Zhang, S. Burer et W. Street : Ensemble pruning via semi-definite programming. *The Journal of Machine Learning Research*, 7 :1315–1338, 2006.

[ZH05] H. Zou et T. Hastie : Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(2) :301–320, 2005.

[ZT03] Z. Zhou et W. Tang : Selective ensemble of decision trees. *In Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, pages 476–483. Springer, 2003.

[ZWT02] Z. Zhou, J. Wu et W. Tang : Ensembling neural networks : many could be better than all. *Artificial intelligence*, 137(1) :239–263, 2002.