université PARIS-SACLAY

Inférence logique de réseaux booléens à partir de connaissances et d'observations de processus de différenciation cellulaire

Logical inference of Boolean networks from knowledge and observations of cellular differentiation processes

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 :
Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Informatique
Graduate School : Informatique et Sciences du Numérique
Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Laboratoire Interdisciplinaire des Sciences du Numérique** (Université Paris-Saclay, CNRS) en collaboration avec **Cancer et génome : Bioinformatique, biostatistiques et épidémiologie des systèmes complexes** (Institut Curie, Université PSL, INSERM), sous la direction de **Christine FROIDEVAUX**, professeure émérite (Université Paris-Saclay, LISN) et les co-encadrements de **Loïc PAULEVÉ**, chargé de recherche HDR (CNRS, LaBRI) et **Andrei ZINOVYEV**, ingénieur de recherche HDR (Institut Curie, U900).

Thèse soutenue à Paris-Saclay, le 29 septembre 2022, par

Stéphanie CHEVALIER

Composition du jury

Simon DE GIVRY Chargé de recherche HDR, INRAE, MIAT (Mathématiques et Informatique Appliquées de Toulouse)	Rapporteur et examinateur
Élisabeth RÉMY Directrice de recherche, CNRS, I2M (Institut de Mathématiques de Marseille)	Rapportrice et examinatrice
Franck DELAPLACE Professeur, Université d'Évry Paris-Saclay, IBISC (Informatique, BioInformatique, Systèmes Complexes)	Examinateur
Carito GUZIOLOWSKI Maîtresse de conférence, Centrale Nantes, LS2N (Laboratoire des Sciences du Numérique de Nantes)	Examinatrice
Fatiha SAÏS Professeure, Université Paris-Saclay, LISN (Laboratoire Interdisciplinaire des Sciences du Numérique)	Examinatrice
Christine FROIDEVAUX Professeure émérite, Université Paris-Saclay, LISN (Laboratoire Interdisciplinaire des Sciences du Numérique)	Directrice de thèse

THESE DE DOCTORAT
NNT : 2022UPASG061

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Motivation et contexte de la modélisation de réseaux de régulation biologique

Sommaire

Construire et simuler un modèle dynamique de la régulation d'un processus biologique est une aide cruciale pour la recherche en biologie et en médecine. Les applications sont nombreuses. Elles vont de l'exploration d'un mécanisme où la construction d'un modèle est, en soi, un outil pour tester des hypothèses et acquérir de nouvelles connaissances, jusqu'à la médecine personnalisée qui nécessite de pouvoir simuler des modèles adaptés aux patients.

Les modèles apportent une description mathématique d'un système biologique, avec les éléments qui le composent et leurs interactions, que ces modèles soient au niveau de la cellule, du tissu, de l'organe, du corps ou de la population. La construction des modèles s'appuie sur les connaissances accumulées, ainsi que sur des données expérimentales dont le nombre a considérablement augmenté ces deux dernières décennies grâce aux immenses progrès des technologies d'observations [Mardis, 2011].

Cependant, il est très difficile de décrire le fonctionnement d'un organisme multicellulaire à partir d'observations expérimentales. Les observations demeurent partielles malgré les progrès technologiques étant donné la très grande complexité des systèmes abordés en biologie. Il est impossible de savoir si l'ensemble des facteurs impliqués dans un mécanisme sont observés. De plus, il ne faut pas négliger la variabilité potentiellement importante entre individus et, pragmatiquement, les observations sont issues de techniques qui ne peuvent être complètement dénuées de biais. Par conséquent, y compris à l'échelle d'une unique cellule, c'est un challenge pour la biologie et la médecine de comprendre suffisamment finement un mécanisme pour créer un modèle mathématique de celui-ci puis, potentiellement, prédire la réaction de ce système face à différentes perturbations.

Ainsi, pour rendre accessible la description mathématique de processus biologiques et améliorer la précision et l'applicabilité de ces modèles, la recherche dépend des progrès non seulement des technologies d'observations, mais également des méthodes d'analyses des observations obtenues. L'objectif est de pouvoir exploiter les connaissances disponibles, bien qu'incomplètes, et pour cela il existe différentes approches de modélisation. Pour la compréhension des mécanismes de régulation des processus cellulaires, la biologie des systèmes porte un intérêt tout particulier aux interactions entre gènes, puisque les gènes sont les composants biologiques portant l'information à l'origine du fonctionnement de la cellule. Comprendre les interactions entre les gènes en créant des modèles du réseau de leurs interactions est donc un besoin majeur. Cette modélisation est utilisée en recherche fondamentale en biologie pour progresser dans la compréhension d'un processus, que celui-ci soit naturel ou provoqué expérimentalement et qu'il soit physiologique ou pathologique pour l'organisme étudié, mais elle aide également la recherche clinique pour la découverte de pistes thérapeutiques. Elle est reconnue comme l'une des solutions les plus prometteuses pour parvenir à une médecine personnalisée, étant donné sa capacité à prédire les effets des médicaments sans avoir à recourir à des expériences *in vivo* ou *in vitro*, avec l'avantage supplémentaire d'accroître l'efficacité tout en réduisant les coûts.

La difficulté de la création de modèles, à partir des connaissances et des données à disposition, limite actuellement de façon importante les applications possibles. Tout particulièrement lorsque sont abordés des comportements cellulaires complexes, mais stratégiques, comme l'acquisition par une cellule de nouvelles fonctions, que ce soit dans le cadre du développement de tissus fonctionnels (embryogenèse, renouvellement cellulaire) ou dysfonctionnels (cancer). La modélisation est encore une tâche lente et fastidieuse, principalement manuelle, qui nécessite une grande expertise sur le système à modéliser. Et malgré toute l'attention portée à cette tâche, il est difficile d'éviter une part de choix arbitraires qui viennent biaiser les résultats obtenus. Pour promouvoir le développement des applications de la modélisation, il faut donc faciliter la démarche en développant les méthodes d'inférence de modèles et limiter les biais de ces modèles. Il faut également que les méthodes proposées puissent s'adapter à la complexité des comportements observés en biologie tout en étant capable de considérer un nombre important de composants, afin de répondre aux besoins pratiques des biologistes.

Les travaux réalisés au cours de ma thèse ont été motivés par ce besoin. L'objectif a été de rendre possible une inférence automatique de modèles de réseaux de régulations, en considérant les observations et les connaissances que les biologistes ont à disposition lorsqu'ils étudient un comportement biologique, afin de modéliser une grande diversité de comportements biologiques potentiellement complexes. Ainsi, les observations exploitées peuvent être issues par exemple du suivi d'une différenciation cellulaire, c'est-à-dire de l'apparition de types cellulaires différents, ou du suivi de perturbations (par exemple l'évolution cellulaire suite à des mutations, l'administration d'un médicament, un changement de conditions environnementales, etc). Afin d'aborder ce défi, j'ai tout d'abord apporté une formalisation du problème, en caractérisant les données à disposition et les comportements biologiques à modéliser pour répondre aux besoins. Sur cette base, j'ai pu développer des parties clés d'un outil d'inférence automatique de modèles afin de garantir la modélisation de comportements complexes tels que des différenciations cellulaires ou des données de perturbations. Puis, afin de communiquer sur les capacités de l'outil, j'ai participé à plusieurs applications de modélisation en garantissant l'inférence de modèles reproduisant des comportements jusqu'à présent non modélisables.

Plan du chapitre

Les données à disposition sur lesquelles baser la création des modèles sont, d'une part, des observations sur le comportement biologique à modéliser avec tout particulièrement des données génomiques, c'est-à-dire des données sur les gènes contenus dans l'ADN d'une ou plusieurs cellules et qui y sont ou non exprimés, et d'autre part, des connaissances sur le système ayant donné ce comportement, connaissances mises à disposition dans des bases de données publiques et qui peuvent être complétées par l'expertise des modélisateurs.

Les données génomiques sont particulièrement exploitées puisque, en étant porteur de l'information à l'origine

des composants biologiques fonctionnels de la cellule, le génome d'une cellule est un acteur central de son fonctionnement. Du fait de ses produits directs (dont l'ensemble est appelé le transcriptome d'une cellule), il régule les processus biologiques qui peuvent tout autant être sains que pathologiques pour l'organisme. Les informations récoltées sur le génome d'une cellule et sur les produits de ses gènes concernent donc des acteurs au cœur du circuit de régulation d'un processus biologique. Afin de comprendre le contexte biologique qui est abordé, je présente dans la première partie de ce chapitre, d'une part, ce que sont le génome et le transcriptome d'une cellule ainsi que les données qui peuvent être collectées pour les observer au cours d'un processus biologique et, d'autre part, ce à quoi correspond le processus de différenciation cellulaire qui, par l'impossibilité jusqu'à présent d'en créer des modèles, a été la motivation initiale des travaux de thèse pour étendre la modélisation à un ensemble de comportements complexes.

Afin de créer des modèles de réseaux de régulations d'un processus biologique, différents formalismes peuvent être utilisés selon les données à disposition et le besoin motivant la modélisation. Mes travaux de thèse abordent la modélisation dynamique de réseaux de régulations via le formalisme logique des réseaux booléens, formalisme applicable pour faire face au manque de données précises, quantitatives et cinétiques, ce qui est généralement le cas pour les grands réseaux de régulations. En effet, malgré son haut niveau d'abstraction, un réseau booléen permet de récupérer les propriétés dynamiques essentielles des systèmes modélisés, ce qui en fait un soutien pour l'exploration de nombreuses problématiques biologiques. La modélisation booléenne se développe particulièrement dans le cadre de recherches liées au cancer (intégrer des données spécifiques aux patients dans la modélisation de la leucémie myéloïde aiguë [Palma et al., 2021], déduire un pronostic de la modélisation d'un type de cancer du sein [Font-Clos et al., 2021a], identifier des cibles thérapeutiques grâce à la modélisation contre le neuroblastome [Dahlhaus et al., 2016], etc). Ces études abordent des questions assez diverses, mais elles concernent essentiellement l'existence d'attracteurs et leurs propriétés d'atteignabilité, sous l'effet de mutations ou de modifications de conditions environnementales, implémentées dans les modèles logiques par une modification des fonctions logiques de régulation. Afin de comprendre le cadre de la modélisation, je présente au sein de la seconde partie de ce chapitre les modèles de réseaux de régulations biologiques qui peuvent être par nature *statiques* ou *dynamiques*, ainsi que de la démarche d'inférence des modèles dynamiques.

Enfin, ce chapitre introductif se termine par la présentation de mes contributions au sein du défi que représente la création de modèles dynamiques de réseaux de régulations.

1.1 Des interactions géniques aux processus biologiques

1.1.1 Le génome et le transcriptome

Les évolutions technologiques de ces deux dernières décennies ont radicalement transformé l'accès aux données biologiques [van Dijk et al., 2014]. En particulier, le séquençage des macromolécules d'ADN (acide désoxyribonucléique) et d'ARN (acide ribonucléique), qui constituait une prouesse technologique il y a 20 ans, est désormais réalisé couramment et a de nombreuses applications en clinique comme en recherche fondamentale (évaluation de la prédisposition à une maladie [Hamdi et al., 2018], reconstruction d'un arbre phylogénétique [Navarro and Martínez-Murcia, 2018], étude d'un comportement cellulaire [Nestorowa et al., 2016], ...) grâce au développement conjoint d'outils bioinformatiques permettant d'analyser ces données [Manzoni et al., 2016].

Le séquençage donne en effet accès à de précieuses informations. ADN et ARN sont complémentaires dans une cellule : la molécule d'ADN porte l'information génétique, appelé génome, et les ARN sont les produits directs de la lecture de l'ADN. Connaître l'enchaînement exact des nucléotides qui composent ces macromolécules biologiques permet ainsi d'identifier d'une part le génome (porté par l'ADN) d'une cellule ou d'une population cellulaire, mais également l'expression de ce génome appelé transcriptome (c'est-à-dire l'ensemble des ARN issus d'un génome). Obtenues dans différents contextes biologiques (conditions expérimentales, types cellulaires, individus, ...), les données génomiques et transcriptomiques offrent à la recherche fondamentale la possibilité d'étudier les variations du génome (séquençage ADN) ou de l'expression du génome (séquençage ARN). La nature et les caractéristiques de ces données vont dépendre de la problématique biologique étudiée et de la technique employée pour réaliser les observations. Afin d'introduire les données utilisées pour la modélisation, je vais brièvement présenter les technologies d'observation du génome et de son expression.

1.1.1.1 Données génomiques

Séquençage d'ADN (*DNA-seq*) Le séquençage d'ADN consiste à déterminer l'ordre dans lequel s'enchaînent les quatre nucléotides possibles (A, T, C, G) composant un fragment d'ADN. C'est en quelque sorte une "lecture" de la molécule d'ADN qui est un très long mot écrit dans un alphabet de quatre "lettres". Les premières méthodes de séquençage ont été développées à la fin des années 1970. La méthode Sanger, la plus utilisée, a notamment permis le séquençage du génome humain grâce à un projet scientifique international de plus dix ans qui a coûté près de trois milliards de dollars : le *Human Genome Project* [Consortium, 2004]. Des progrès technologiques remarquables ont depuis continué de révolutionner la génomique et la biologie de manière générale, tout particulièrement l'avènement du séquençage dit à *haut-débit*, également appelé séquençage *de nouvelle génération* (NGS, pour *Next Generation Sequencing*). Cette technologie a considérablement facilité l'accès au séquençage, en diminuant son coût tout en augmentant sa rapidité. Il est ainsi désormais possible de séquencer le génome humain en quelques jours

pour moins de 1000 dollars. Grâce à cette évolution, les données NGS sont aujourd'hui produites en masse pour des problématiques biologiques diverses afin, par exemple, d'analyser les variations génétiques, rechercher des marqueurs d'une maladie, reconstruire des histoires évolutives (arbres phylogénétiques), etc. La figure 1.1 montre l'évolution du coût du séquençage d'un fragment de 1000 nucléotides, de 2001 à 2021.

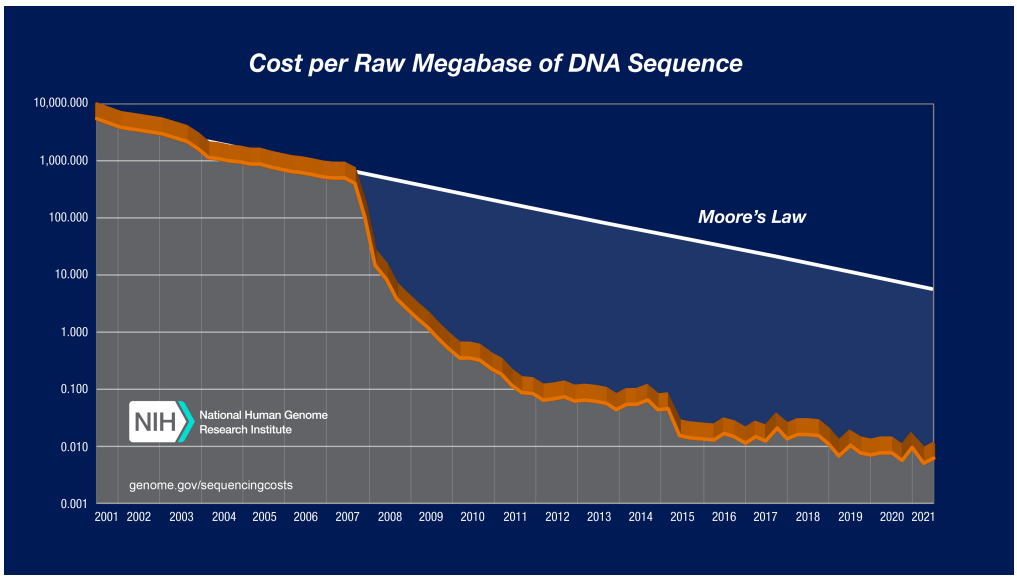


FIGURE 1.1 – Évolution du coût du séquençage d'un fragment d'ADN de 1000 nucléotides (en USD), de 2001 à 2021. Source : https://www.genome.gov/sequencingcosts/

Puces à ADN (*DNA microarray*) Cette technique permet de rechercher la présence de séquences particulières dans le génome. Elle provient d'une méthodologie présentée en 1975 [Grunstein and Hogness, 1975] progressivement améliorée jusqu'à ce qu'elle apparaisse pour la première fois sous son nom actuel dans une publication de 1995 [Schena et al., 1995]. Elle fonctionne sur le principe de l'hybridation entre deux brins d'ADN, c'est-à-dire la propriété d'association qu'ont deux brins d'ADN dits complémentaires afin de former une double hélice. Des simples brins d'ADN, appelés sondes, sont accrochés à un support qui est mis en contact avec les portions du génome à analyser, appelées cibles. L'hybridation de cibles au support informent de la présence de séquences correspondant aux sondes au sein du génome étudié. Il est ainsi possible de tester la présence de telles ou telles variations et de comparer ces résultats d'hybridation entre plusieurs génomes.

Séquençage de chromatine immuno-précipitée (*ChIP-seq*) Cette technique, mentionnée dans des publications à partir de 2007 [Robertson et al., 2007], permet de connaître les séquences nucléotidiques où se lie une protéine d'intérêt. En identifiant ainsi les sites de liaison d'une protéine dans le génome, on détermine les gènes ciblés par cette protéine et ainsi des liens d'influence entre gènes, de celui qui code une protéine à celui qui est ciblé par cette protéine. La technique ChIP-seq consiste à isoler les séquences d'ADN qui se lient à la protéine d'intérêt puis à identifier ces séquences, soit via une puce à ADN soit via un séquençage.

1.1.1.2 Données transcriptomiques

La connaissance assez précise du génome obtenue grâce au séquençage d'ADN a ouvert la voie à l'analyse de l'expression des gènes. Ce sont tout d'abord les micropuces qui ont permis cette révolution, technique désormais complétée par le séquençage du transcriptome plus sensible et non limité à la détection d'un ensemble fini d'ARN pour lesquels une sonde a été prévue.

Puces à ARN (*RNA microarray*) Puces tout à fait similaires aux puces à ADN, elles reposent sur l'hybridation entre brins d'ADN complémentaires. La différence est qu'ici le support est mis en contact avec les ADN complémentaires des ARN contenus dans l'échantillon à analyser. La puce détecte ainsi la présence de séquences particulières dans le transcriptome, permettant d'obtenir un niveau relatif d'expression de gènes entre différentes conditions expérimentales. Le terme puces à ARN désigne donc les puces utilisées pour comparer les niveaux d'ARN présents dans un milieu biologique, tandis que les puces à ADN identifient la séquence ADN ou les niveaux d'ADN. Les puces à ARN sont largement utilisées pour comparer les niveaux d'expression de gènes entre plusieurs conditions biologiques car il est possible, à partir de ce type de données, de déduire des mécanismes de régulation entre gènes. Pour davantage d'informations, le papier [Sealfon and Chu, 2011] donne un aperçu de la méthode et des applications des puces à ARN et ADN dans divers domaines de la recherche biologique.

Séquençage d'ARN (*RNA-seq*) L'ARN étant complémentaire de l'ADN, on peut comparer sa séquence avec celle de la molécule d'ADN pour connaître le gène dont l'ARN est le produit. Selon la quantité d'ARN séquencé correspondant à un gène, on peut déduire le niveau d'expression de ce gène à un instant t et voir quelles sont les parties du génome les plus actives en fonction de différentes conditions biologiques et expérimentales. Comme la puce à ARN, le RNA-seq est de ce fait majoritairement utilisé pour comparer les niveaux d'expression des gènes entre des cellules à différents points de temps d'une expérience ou entre cellules soumises à des conditions expérimentales différentes. Mais contrairement à la puce, la recherche des transcrits ne se limite pas aux séquences ARN déterminées à l'avance et fixées sur la puce ; il est ainsi possible d'identifier des transcrits sans les connaître à l'avance.

Le progrès des techniques de séquençage à haut-débit a permis l'apparition des méthodes de séquençage au niveau de la cellule [Shapiro et al., 2013], séquençage communément appelé *single-cell*, avec le RNA-seq comme application majeure (scRNA-seq). Il existe de ce fait deux échelles de séquençage du transcriptome. Les méthodes classiques de RNA-seq permettent d'obtenir l'expression moyenne des gènes au sein d'une population cellulaire en séquençant les ARN de cette population (séquençage *bulk*), résultant en une moyenne des profils transcriptomiques de toutes les cellules d'une population. À l'inverse, le séquençage transcriptomique *single-cell* permet de séquencer cellule par cellule, ce qui a l'avantage de montrer la diversité des profils transcriptomiques des cellules d'une population. J'apporte ci-dessous des précisions sur les applications et analyses usuelles de ces deux types de séquençage.

	obs1	obs2	obs3	obs4	obs5
gene1	1	2	3	0	0
gene2	0	0	0	0	0
gene3	9	51	3	46	56
gene4	0	0	153	0	0
gene5	73	149	130	15	346
gene6	0	0	0	0	0

Tableau 1.1 – Extrait d'une table de comptage ARN, avec les valeurs de comptage ARN mises en lien avec 6 gènes différents au sein de 5 observations.

Le séquençage *bulk* est utilisé pour comparer les profils d'expression de plusieurs échantillons, souvent collectés selon deux modalités d'échantillonnage en fonction de la problématique. Une première manière d'échantillonner consiste à prélever au sein de plusieurs conditions biologiques ou expérimentales afin d'analyser les différences d'expression entre ces conditions. On peut ainsi étudier, par exemple, l'impact de perturbations telles que des mutations ou l'administration de médicaments sur l'expression des gènes. Lorsque l'objectif est par contre d'analyser l'évolution de l'expression des gènes au cours d'un phénomène biologique ou d'une expérimentation, les échantillons à séquencer sont collectés à différents points de temps. Les données d'expression forment alors une série temporelle d'observations. Le tableau 1.1 illustre une table de comptage ARN au sein de plusieurs observations pouvant correspondre à différents points de temps d'une expérience ou à différentes conditions expérimentales.

L'avancée technologique du **séquençage ARN *single-cell (scRNA-seq)*** permet d'observer l'expression des gènes à l'échelle de la cellule et donc l'hétérogénéité du transcriptome au sein d'une population cellulaire. Des méthodes statistiques peuvent s'appliquer sur ces données pour, par exemple, mettre en évidence différents types de cellules au sein de l'échantillon. Lorsqu'on s'intéresse à une population cellulaire en différenciation, des méthodes permettent également de reconstruire une trajectoire évolutive entre les cellules, appelée *pseudo-temps*. Un biais du séquençage à l'échelle de la cellule est sa moindre sensibilité par rapport au séquençage à l'échelle de la population, si bien que des gènes faiblement exprimés peuvent être considérés comme non exprimés.

1.1.1.3 Apport de ces données pour l'étude des processus biologiques

L'accès à l'information génétique et à son expression au sein de différents contextes biologiques et expérimentaux est précieux pour comprendre les origines des comportements biologiques qu'on observe. Le projet de la thèse est né du besoin d'élucider des mécanismes qui sous-tendent le développement et la spécialisation cellulaire. Ces processus d'évolution des cellules impliquent des événements appelés *bifurcations*, au cours desquels des cellules de distinguent progressivement de leur type cellulaire initial pour évoluer vers des destins différents, tels qu'une spécialisation cellulaire ou un état pathologique. Déterminer les composants biologiques régulateurs de ces bifurcations est un enjeu majeur de nombreuses recherches pour étendre les connaissances et envisager de nouvelles pistes thérapeutiques. En vue d'une méthode pour aider à mettre en évidence les mécanismes génétiques qui permettent à une cellule d'acquérir de nouvelles fonctions, nous nous sommes donc tout particulièrement

intéressés aux phénomènes de bifurcations dans les processus biologiques.

1.1.2 La différenciation cellulaire

La différenciation cellulaire est le processus au cours duquel une cellule évolue d'un type cellulaire à un autre, en acquérant des compétences et des caractéristiques la distinguant irréversiblement de son type d'origine.

1.1.2.1 Différenciations saines

La différenciation cellulaire est indispensable au développement d'un organisme pluricellulaire afin qu'à partir de quelques cellules se forment ses différents tissus. Ce phénomène se poursuit ensuite au sein de certains tissus afin qu'un renouvellement cellulaire le maintienne en état fonctionnel (sang, paroi intestinale et épiderme notamment). Les cellules physiologiquement capables de différenciation le sont à différents niveaux. Les cellules aux capacités de différenciation les plus élevées sont appelées totipotentes et pluripotentes et n'existent qu'au cours de l'embryogenèse. Elles peuvent se différencier en toutes les cellules possibles de l'organisme auquel elles appartiennent, les totipotentes étant également capables de former le placenta. Dans un organisme adulte, on observe différents grades de cellules capables de différenciation allant des cellules multipotentes capables de former différents types cellulaires relativement proches, à des cellules unipotentes se différenciant en un seul type cellulaire en plus d'assurer leur propre renouvellement.

La figure 1.2 illustre ce phénomène sain de différenciation cellulaire avec le renouvellement des cellules du sang à partir des cellules souches hématopoïétiques présentes dans la moelle osseuse. Les évolutions possibles de types cellulaires sont représentées sous la forme d'un arbre : à la racine sont placées les cellules souches hématopoïétiques, c'est-à-dire les cellules primitives à l'origine de toutes les lignées de cellules sanguines du corps, tandis que les feuilles représentent quatre spécialisations possibles. Les branches de l'arbre indiquent les différents types cellulaires intermédiaires par lesquels passe une cellule ainsi que les événements de bifurcations entre les différents types, étapes clés du phénomène de différenciation avec l'acquisition irréversible de compétences et caractéristiques distinguant les familles de cellules. Une fois que la cellule est différenciée, elle ne peut pas naturellement se transdifférencier en un autre type cellulaire pourtant issu de la même cellule souche (un lymphocyte T ne peut pas naturellement se transformer en un granulocyte) : la bifurcation est irréversible.

1.1.2.2 Différenciations pathologiques

Le processus sain de différenciation cellulaire est entraîné par une adaptation épigénétique, c'est-à-dire une modification non pas du génome mais de son expression pour aboutir à des cellules de types distincts. Mais une transformation cellulaire peut également être le résultat de mutations génétiques ou de composants perturbateurs du fonctionnement des gènes. Ainsi le cancer est la conséquence d'une différenciation pathologique, avec des cellules

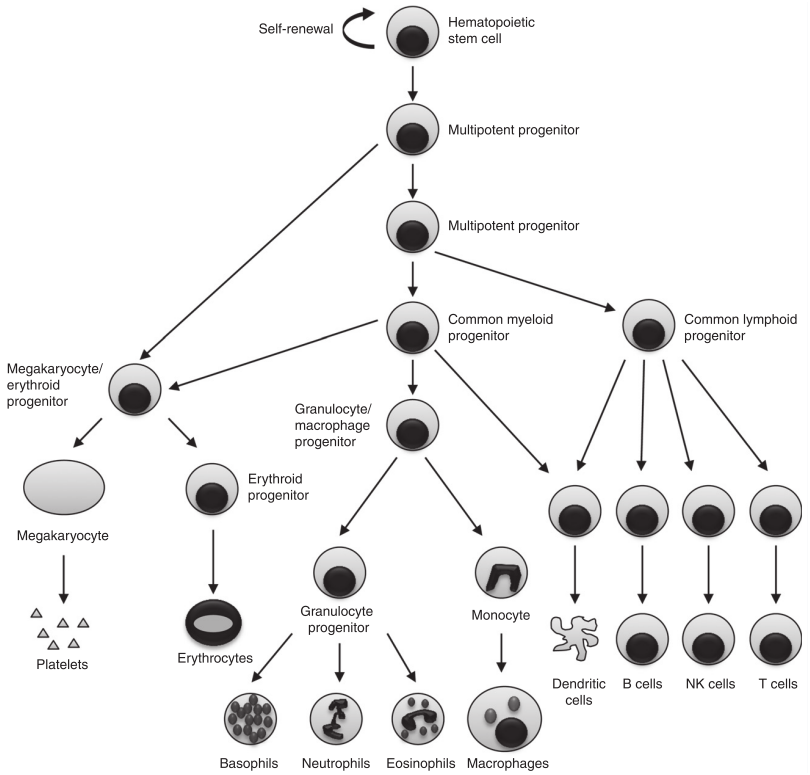


FIGURE 1.2 – Schéma des différenciations possibles à partir d'une cellule souche hématopoïétique [Hoggatt and Pelus, 2013]

acquérant de nouvelles compétences et caractéristiques malheureusement nuisibles au bon fonctionnement de l'organisme.

1.1.2.3 Différenciations provoquées expérimentalement

Des différenciations cellulaires sont provoquées en laboratoire en soumettant des cellules à différentes conditions expérimentales (mutations, perturbations chimiques ou environnementales) afin d'observer leur évolution, tel qu'illustré par la figure 1.3. Ces expérimentations sont utilisées pour étudier les processus sains et pathologiques de différenciation, mais également pour tenter de forcer des différenciations inhabituelles à des fins d'exploration du processus et de solutions thérapeutiques pour réparer des tissus lésés par exemple [Jopling et al., 2011].

1.1.2.4 Nature des observations

Les mécanismes régissant les phénomènes de différenciation sont étudiés au travers de l'évolution des expressions de gènes. Cette évolution d'expression peut être observée en *bulk RNA-seq* (cf. section 1.1.1.2) via des mesures à différents points de temps et/ou dans plusieurs conditions expérimentales en vue de les comparer. Elle peut également l'être avec des données *single-cell RNA-seq* issues dans ce cas d'un unique prélèvement. En *single-cell*, ce prélèvement donne le panel des profils d'expressions des cellules mais sans fournir, étant donné sa nature, un ordre entre ces profils d'expression. Cependant, via l'analyse des profils d'expression des différentes

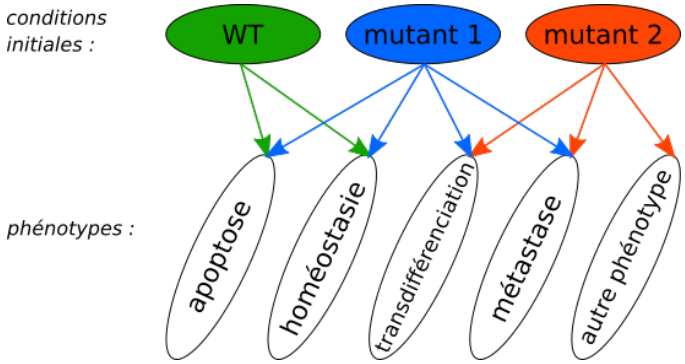


FIGURE 1.3 – Représentation schématique et hypothétique des destins cellulaires observés à partir de trois conditions initiales (cellules non perturbées dites *Wild Type* (WT) et cellules soumises à deux mutations distinctes)

cellules présentes dans l'échantillon, des méthodes dites de reconstruction de trajectoire permettent d'inférer un *pseudo-temps*. À partir d'un prélèvement *single-cell* qui contient, par exemple, des cellules en cours de différenciation, il est possible d'étudier l'évolution de l'expression des gènes au cours d'un processus biologique une fois que les cellules ont été organisées le long d'un pseudo-temps retraçant le processus de différenciation.

Reconstruction de trajectoire à partir de données *single-cell* À partir d'un séquençage transcriptomique *single-cell* (cf. paragraphe en 1.1.1.2), un algorithme dit de reconstruction de trajectoire a pour objectif d'ordonner les cellules afin de former une trajectoire appelée *pseudo-temps*. Cette trajectoire reconstruite met en lumière le processus évolutif expliquant l'hétérogénéité des cellules de l'échantillon, cellules en cours de différenciation par exemple ou engagées dans un autre type de transition biologique. Une méthode de reconstruction de trajectoire offre ainsi la possibilité de suivre l'évolution des expressions de gènes au cours d'une transition cellulaire. Différents algorithmes de reconstruction de trajectoire sont actuellement proposés, notamment STREAM [Chen et al., 2019] et Monocle2 [Qiu et al., 2017] qui sont deux méthodes reconnues et couramment utilisées. Ces méthodes reposent sur une réduction de la dimensionnalité des données, mettant en avant la proximité des cellules selon leur expression. Une structure d'arbre est alors apprise pour classifier et ordonner les cellules dans l'espace (figure 1.4). En définissant (manuellement) la racine, on définit l'orientation des trajectoires et des bifurcations.

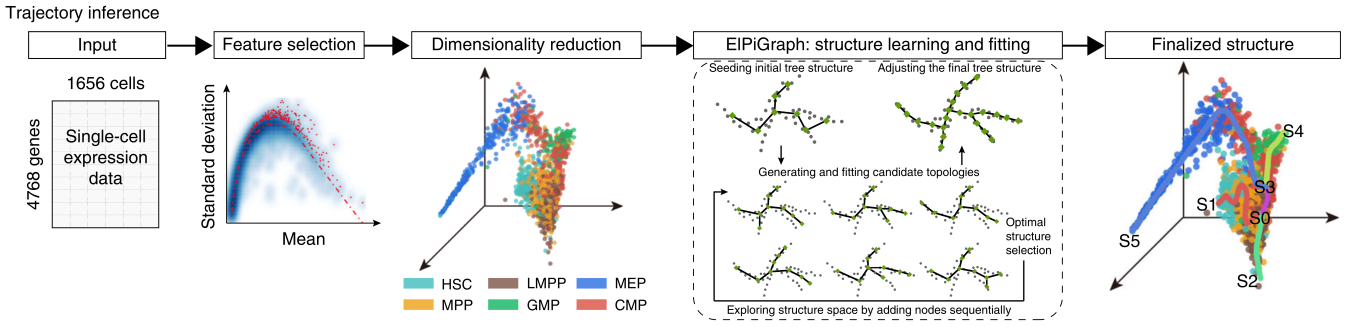


FIGURE 1.4 – Illustration de la méthodologie suivie pour la reconstruction de trajectoire [Chen et al., 2019]

Les trajectoires inférées peuvent être visualisées sous la forme d'un arbre tel qu'illustré en figure 1.5 sur lequel chaque point représente une cellule. Les cellules y sont coloriées en fonction du type cellulaire auquel elles appartiennent (déterminé à partir de l'expression de marqueurs manuellement définis) et l'abscisse correspond au pseudo-temps. Les extrémités des branches sont identifiées par un label ; S1 a été choisie comme racine puisqu'elle concentre les cellules multipotentes tandis que S2, S4 et S5 concentrent les cellules différenciées. La dispersion des points autour des branches reflète leur distance dans l'espace réduit.

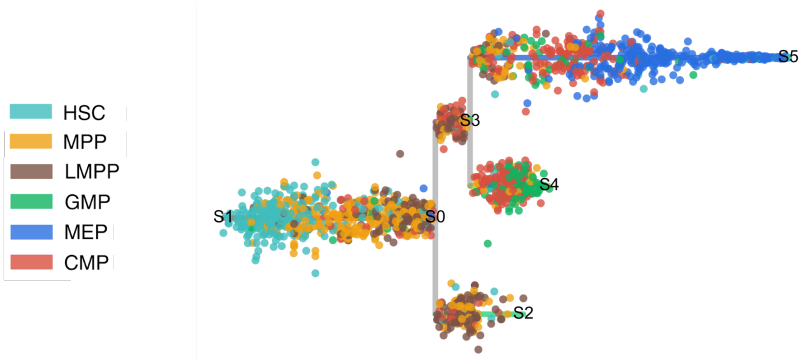


FIGURE 1.5 – Trajectoire de différenciation cellulaire reconstruite à partir de données *single-cell* via STREAM.

1.2 Modèles de réseaux de régulation biologique

Pour aider à comprendre le fonctionnement d'un système complexe tel que celui des réseaux de régulation, les biologistes utilisent des outils et concepts mathématiques pour le décrire et ainsi réaliser un modèle du système observé. Un modèle constitue une représentation restreinte et simplifiée de la réalité qui éclaire la compréhension, guide la poursuite des expérimentations et peut ouvrir la voie à la réalisation de prédictions afin d'évaluer l'impact de perturbations sur le système (par exemple l'administration d'un médicament, la mutation d'un gène...).

1.2.1 Modèles statiques

L'analyse des données issues de *ChIP-seq*, *microarray* ou de séquençage à haut débit permet de mettre en évidence des influences entre gènes via leurs produits (ARN, protéines). Concernant des gènes en lien avec la différenciation des cellules du sang (hématopoïèse), une étude [Koh et al., 2013] a par exemple montré que la protéine issue de l'expression du gène RUNX1 favorise l'expression d'un autre gène, nommé SPI1. L'expression de SPI1 est également favorisée par la protéine CEBPA (produit du gène homonyme) [Yeamans et al., 2007], tandis que la protéine du gène GFI1 bloque la protéine de SPI1, inhibant sa fonction [Dahl et al., 2007]. Un gène est dit activateur ou inhibiteur d'un autre gène lorsque son produit agit positivement ou négativement soit sur l'expression du second gène en favorisant ou inhibant son expression, soit sur le produit de ce second gène en participant à sa fonction ou en la bloquant. Ainsi, RUNX1 et CEBPA sont des activateurs de SPI1, tandis que GFI1 est un inhibiteur

de SPI1. Ces connaissances peuvent être synthétisées par des liens d'influences entre gènes au sein d'un graphe.

1.2.1.1 Graphe d'interactions

Les influences mises en évidence entre gènes peuvent être représentées par un graphe d'interactions (aussi appelé graphe d'influence) tel qu'illustré en figure 1.6. Les nœuds du graphe correspondent aux gènes et les arcs représentent leurs interactions. Les arcs sont étiquetés : un arc est positif (en vert sur la figure) si sa source est un activateur de sa cible et négatif (en rouge) dans le cas d'un inhibiteur. Les connaissances accumulées sur les influences entre gènes sont ainsi modélisées sous la forme de graphes d'interactions au sein de bases de données d'interactions.

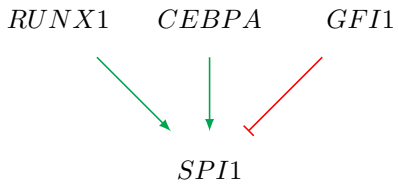


FIGURE 1.6 – Extrait d'un graphe d'interactions

1.2.1.2 Bases de données d'interactions

La mise en place de bases de données thématiques a accompagné l'augmentation massive du nombre de données biologiques, facilitant le partage des données et de connaissances. Des bases de données d'interactions telles que DoRothEA [Garcia-Alonso et al., 2019] et Signor [Licata et al., 2019] ont ainsi été conçues et sont maintenues à jour pour regrouper les connaissances acquises sur les interactions entre composants biologiques. Les influences entre gènes citées précédemment en exemple ont été extraites de la base de données Signor.

Lors de l'utilisation de ces bases de données, il faut garder à l'esprit qu'elles rassemblent des influences mises en évidence par des études différentes dans des contextes expérimentaux différents, ce qui ne permet pas de conclure directement sur la pertinence de ces influences pour une problématique biologique donnée. Il est ainsi difficile de postuler qu'une interaction entre deux gènes mise en évidence au sein de cellules souches du sang a également lieu dans des cellules extraites d'une tumeur pancréatique. D'autant plus que l'observation d'un système biologique permet rarement de connaître l'effet de la combinaison de plusieurs acteurs. Ainsi, l'information "RUNX1 est un activateur de SPI1" ne permet pas de savoir si l'expression de RUNX1 est suffisante pour activer SPI1 ou si elle nécessite la coopération d'un ou plusieurs autres gènes pour lesquels une influence sur SPI1 a également été mise en évidence, tels que CEBPA et GFI1. Le graphe d'interactions entre gènes d'intérêt extrait d'une base de données d'interactions constitue donc une base de connaissances qu'il est nécessaire de confronter à une expertise et à des observations sur le phénomène biologique étudié.

1.2.1.3 Extraire des liens entre gènes à partir des données d'expression

Il est possible de déterminer des corrélations d'expression entre gènes à partir des données d'expression génique. Différentes méthodes existent suivant que ces données soient issues de *microarray* (ARACNE [Margolin et al., 2006], MIIC [Cabeli et al., 2020]) ou de séquençage d'ARN (TimeDelay-ARACNE [Zoppoli et al., 2010]). L'analyse de ces co-expressions permet parfois d'aboutir à des liens causaux. On peut par exemple supposer que l'activité de facteurs de transcription est à l'origine de l'expression de ses cibles. Inférer ces causalités à partir des données d'expression offre la possibilité de construire des réseaux causaux directement à partir des données collectées sur le système étudié. Cependant, au-delà de la difficulté de conclure sur les causalités limitant l'obtention d'un réseau assez large pour être représentatif des gènes impliqués, ce réseau est représentatif uniquement d'interactions ayant été observées et est donc insuffisant pour prédire des comportements non-observés. Inférer des interactions à partir des données d'expression constitue donc une opportunité de compléter les connaissances fournies par les bases de données, en y ajoutant des interactions directement en lien avec le système étudié.

1.2.1.4 Limite des modèles statiques

La notion d'états d'expression des gènes qui évoluent dans le temps est absente des graphes d'interactions. Pour reprendre notre exemple d'interactions présenté en figure 1.6, les connaissances dont nous disposons ne répondent pas aux questions suivantes : RUNX1 est-il suffisant pour activer SPI1, ou faut-il que CEBPA le soit en même temps ? Que se passe-t-il si GFI1 est également exprimé ? Parmi les nombreuses interactions mises en évidence et réunies au sein d'un graphe d'interactions, rares sont les informations sur la combinaison de ces interactions sur une même cible. Pourtant, les phénomènes biologiques observés résultent de ces combinaisons. C'est un enjeu majeur pour la biologie de les révéler.

1.2.2 Modèles dynamiques

Certaines activations et inhibitions représentées dans un modèle statique ne sont effectives que pour des profils particuliers d'expression de gènes. Par exemple, l'inhibition de GFI1 sur SPI1 représentée en figure 1.6 est peut-être effective uniquement lorsque CEBPA n'est pas exprimé. Pour progresser dans la compréhension d'un processus biologique, il est de ce fait nécessaire d'élucider comment les différents influenceurs d'une même cible combinent leurs effets pour aboutir au comportement observé. À cette fin, la création d'un modèle dynamique de réseau de régulation génique raffine un modèle statique en prenant en compte cette fois le contexte d'activité des gènes : l'état d'expression des gènes et l'évolution temporelle de ces états d'expression. Suivant le formalisme choisi, cette évolution temporelle peut prendre la forme d'un temps logique avec une succession d'états d'expression, ou d'un temps chronométrique continu ou discret. La modélisation peut être quantitative avec des variables mesurables (concentration, nombre de molécules à un instant donné, ...) soumises à des paramètres de pondération pour

représenter l'évolution de ces variables au cours du temps, souvent décrite sous forme d'équation différentielle ordinaire (EDO). Elle peut également être qualitative avec l'utilisation de paramètres d'opérateurs logiques (comme *"et"* $\wedge$, *"ou"* $\vee$) entre les variables qui, au lieu d'être des valeurs mesurables telles que la concentration, seront une interprétation des mesures indiquant simplement si la molécule est présente en nombre suffisant ou non pour agir. Le choix du formalisme dépend du contexte et des besoins. Une modélisation quantitative, sous la forme d'EDO, permet d'aborder l'évolution des concentrations des produits d'une voie métabolique lors d'une augmentation en glucose par exemple. La principale limite de l'application des EDO en biologie est la nécessité de déterminer les paramètres de la dynamique des interactions entre les éléments. Il est en effet fréquent de ne pas disposer de suffisamment de connaissances pour paramétrer les interactions. Une modélisation qualitative, sous la forme de réseau booléen, est quant à elle adaptée à la précision des données biologiques sur des phénomènes à l'échelle de la cellule tels que la détermination des destins cellulaires (*cell fate decision*). Plus intelligible, ce type de formalisme permet une conception et une mise en application directe à partir des données.

1.2.2.1 Choix d'une modélisation logique booléenne

Parmi les nombreux paradigmes de modélisation des réseaux d'interactions, le formalisme de réseau booléen [Thomas, 1973] permet de modéliser les interactions dans le contexte de systèmes peu spécifiés en capturant un comportement biologique tout en nécessitant peu de paramètres par rapport à un modèle quantitatif. Les réseaux booléens (RBs), et les modèles logiques en général, sont ainsi largement adoptés pour la modélisation des réseaux de signalisation et de réseaux de gènes et facteurs de transcription car ils sont adaptés à l'échelle et à la précision des données biologiques sur les interactions moléculaires. Ils sont assez proches de la granularité des connaissances actuelles en biologie.

Les composants d'un réseau booléen, qui représentent des composants biologiques tels que des gènes, des protéines, des conditions environnementales ou encore des comportements cellulaires, peuvent être actifs ou inactifs et les interactions entre composants sont décrites par des règles logiques indiquant l'état de chaque composant en fonction de celui de ses influenceurs. C'est un haut niveau d'abstraction qui décrit de façon très intelligible les interactions biologiques, et qui peut malgré cela reproduire des dynamiques complexes incluant un nombre important de composants. De plus, un réseau booléen ne dépend pas de données quantitatives précises qui sont souvent hors de portée des observations recueillies. Ainsi, à partir des données disponibles, un réseau booléen peut être le modèle d'une dynamique biologique complexe, facilement intelligible pour l'interprétation et l'étude du processus modélisé. Le chapitre suivant apportera une définition de ce formalisme qui est de plus en plus utilisé pour la modélisation de la décision cellulaire engageant les cellules dans différentes transitions. Je me suis particulièrement intéressée aux cellules du sang, avec notamment les travaux de modélisation de [Collombet et al., 2017] et [Schwab et al., 2021] sur la différenciation et le vieillissement des cellules du sang, mais la modélisation booléenne intéresse globalement pour

tous types de transition cellulaire ([Kaushik and Sahi, 2015] sur le diabète, [Font-Clos et al., 2021b] sur le cancer du sein, [Montagud et al., 2022] sur le cancer de la prostate, etc).

1.2.2.2 Démarche de modélisation

Pour explorer les mécanismes à l'origine des comportements observés, les modélisateurs cherchent à construire des modèles d'interactions entre gènes dont le comportement reproduit un comportement biologique souhaité. Ils souhaitent ainsi déterminer les interactions et règles qui permettent de reproduire l'évolution des observations d'expression des gènes, afin de d'identifier les différents contextes dans lesquels un gène va s'exprimer. Pour cela, ils confrontent d'une part des observations du comportement biologique qu'ils souhaitent modéliser (mesures d'expression de gènes principalement, au cours d'un comportement dont les propriétés dynamiques sont décrites de façon experte par le modélisateur) avec, d'autre part, les connaissances accumulées sur les interactions rassemblées au sein de graphes d'interactions afin d'y rechercher les interactions ayant mené au comportement observé.

Construction experte de modèles d'interactions de gènes En biologie des systèmes, la grande majorité des modèles sont construits par une approche d'essais-erreurs qui impliquent des choix pour préciser les règles du modèle jusqu'à ce que sa dynamique concorde avec le comportement observé. Cette démarche par tâtonnements s'explique par la difficulté d'exploiter les données biologiques en vue de modéliser un comportement cellulaire. Cette difficulté provient d'une part de l'hétérogénéité des sources de connaissances réunies au sein des graphes d'interactions, avec des données provenant d'une multitude d'expériences menées dans des contextes biologiques différents. Ainsi, l'influence d'un gène A sur un gène B peut être issue d'une analyse ChIP-seq sur des cellules pancréatiques saines tandis que l'influence du gène C sur B provient d'une expérimentation sur des cellules du sang soumises à différentes mutations. Il n'est alors pas possible de déterminer directement les interactions à considérer pour le système étudié. D'autre part, les données d'expression à confronter au domaine des connaissances sont partielles même en RNA-seq puisque par biais de séquençage on ne peut pas espérer couvrir l'ensemble des gènes de l'organisme étudié. De plus, il faut tenir compte de l'impact des biais expérimentaux lorsqu'on considère les valeurs d'expression de gènes. La mesure brute ne fait pas sens en soi car la valeur est biaisée par de trop nombreux paramètres (séquence du transcrit, sensibilité de la méthode, biais des manipulations expérimentales), mais elle apporte une information qualitative comparable au sein d'une même expérimentation. La construction de modèles est une tâche qui implique de ce fait une interprétation des observations biologiques.

En raison de ces caractéristiques des connaissances et observations biologiques, la modélisation demeure une démarche encore majoritairement manuelle. Mais elle se confronte à une difficulté d'une autre nature. Considérer les réseaux booléens dont la structure est compatible avec les connaissances sur les interactions et avec la dynamique de données engendre une explosion combinatoire du nombre de règles possibles et donc à un espace rapidement immense du nombre de réseaux booléens candidats. Au-delà des connaissances expertes des modélisateurs, la

définition des règles du modèle reposent donc sur un nombre important de choix arbitraires puisqu'il est impossible de parcourir l'ensemble des réseaux booléens possibles.

Dans ce contexte, confronter d'une part un graphe d'interactions basé sur les connaissances délimitant la structure possible des modèles avec, d'autre part, des observations partielles du système, revient dans la grande majorité des cas à tenter de résoudre un problème largement sous-spécifié qui admet un nombre énorme de modèles candidats. En pratique, les données biologiques laissent la possibilité d'une multitude de réseaux booléens candidats. Par conséquent, des choix arbitraires de modélisation doivent être faits, par exemple en donnant la priorité à certaines logiques entre régulateurs ou en préférant les plus petits/les plus grands modèles, ce qui biaise les prédictions faites ultérieurement avec le modèle. La construction manuelle de modèles par cycle d'essai-erreur afin de rapprocher les simulations des résultats souhaités est de ce fait une démarche fastidieuse et aveugle sur l'espace des modèles possibles, qui nécessite des choix arbitraires dont l'impact est difficile à estimer sur les prédictions basées sur ces modèles.

Synthèse automatique de réseaux booléens En vue de faciliter la création de modèles pour appréhender les phénomènes biologiques complexes, des méthodes de synthèse automatique de réseaux booléens compatibles avec des données biologiques se sont développées. Pour cela, la première stratégie a été de mimer la démarche manuelle d'essai-erreur afin de l'automatiser. Sur la base d'une hypothèse forte sur les interactions telle que requérir au moins un activateur et aucun inhibiteur, les méthodes enchaînent des cycles de modifications progressives des règles avec des simulations du modèle en vue de s'approcher des expressions observées. La méthode de synthèse automatique à laquelle les travaux de ma thèse ont contribué, BoNesis, explore quant à elle l'espace des solutions en s'éloignant de cette démarche. La recherche de modèles est décrite sous la forme d'un problème de satisfiabilité, tenant compte de l'ensemble des connaissances accumulées, de l'expression des gènes observés au cours d'une expérience, ainsi que du comportement du système observé formalisé dans le contexte des réseaux booléens et décrit de façon experte. BoNesis infère automatiquement l'ensemble des modèles constitués uniquement d'interactions connues incluses dans un graphe d'interactions donné, et qui reproduisent les évolutions observées d'expression de gènes selon les comportements formalisés.

L'approche automatique ne fait sens qu'avec une implication manuelle experte. En effet, spécifier et formaliser le problème biologique est indispensable pour obtenir un ensemble de modèles à la fois pertinents et analysables. Ainsi, il faut parvenir d'une part à des modèles pertinents, c'est-à-dire constitués d'interactions connues ou plausibles étant donné le comportement observé, et construit en exploitant des observations dont le traitement nécessite lui-même l'expertise biologique. Et d'autre part, il faut s'assurer d'obtenir des modèles ensuite analysables, en cherchant le compromis entre une taille de modèles à la fois suffisamment grande pour ne pas manquer de composants et d'interactions impliqués, mais également assez restreinte pour mettre en évidence le cœur de la régulation du comportement observé.

1.3 Contributions et plan du manuscrit

Ma thèse est un travail interdisciplinaire réalisé au *Laboratoire Interdisciplinaire des Sciences du Numérique* de l'université Paris-Saclay (*Laboratoire de Recherche en Informatique* jusque fin 2020), en collaboration avec l'unité *Cancer et génome : Bioinformatique, biostatistiques et épidémiologie des systèmes complexes (U900)* de l'institut Curie. Ce projet de recherche, dont les travaux contribuent au développement d'une médecine personnalisée en cancérologie, a été soutenu financièrement par l'*Institut Thématique Multi-Organismes Cancer* (ITMO).

Mes travaux de thèse portent sur la modélisation qualitative des processus biologiques, modèles sous la forme de réseaux booléens dont je présente le formalisme au chapitre 2. L'objectif a été d'étendre les applications possibles, guidée par les besoins exprimés par les modélisateurs en cancérologie et en biologie du développement. Ces deux thématiques focalisent leur attention sur des processus de différenciations cellulaires, qui sont gouvernés par des mécanismes de décision cellulaire dont la compréhension est un enjeu scientifique majeur.

Aborder cette modélisation a nécessité en premier lieu de caractériser les comportements biologiques présentés comme un enjeu de modélisation. Ma première contribution, présentée au chapitre 3, est de fournir un cadre formel permettant de raisonner sur les propriétés dynamiques souhaitées au sein des réseaux booléens. J'ai progressivement détaillé les comportements observés dans les données disponibles pour répondre aux problématiques de modélisation sur les mécanismes de décision cellulaire, afin de formaliser au fur et à mesure les propriétés dynamiques attendues au sein des modèles.

Afin de rendre possible l'inférence de réseaux booléens modèles d'une différenciation cellulaire selon la formalisation établie, un outil d'inférence automatique de modèles, nommé BoNesis, a été développé. Cet outil aborde la modélisation logique comme un problème de satisfiabilité. J'ai participé à son développement et j'ai créé des contraintes en Answer-Set Programming, présentées au chapitre 4, qui garantissent que seuls les réseaux booléens compatibles avec les données biologiques en entrées sont inférés. Afin d'aider le processus de création de modèles, BoNesis (dont je détaille le principe au début du chapitre 5) permet de confronter un vaste ensemble de connaissances sur les interactions, tels que ceux disponibles dans les bases de données d'interactions biologiques, avec une combinaison d'observations expérimentales de différentes natures. L'outil étend ainsi l'inférence de modèles à des comportements qui n'étaient pas abordables jusqu'à présent, tout en explorant l'espace des modèles pour garantir l'inférence de toutes les solutions. BoNesis recouvre de ce fait les méthodes d'inférence abordant les états stables et les trajectoires tout en permettant la modélisation de comportements biologiques plus complexes. De plus, il passe à l'échelle des grands réseaux d'interactions tout en proposant une méthodologie pour centrer le modèle sur les composants d'intérêt étant donné les données expérimentales fournies. Enfin, en permettant l'obtention de l'ensemble des modèles possibles (ou d'un sous-ensemble diversifié de ces modèles), il aide la mise en évidence des interactions sur lesquelles repose le comportement observé, analyse cruciale dans le contexte des réseaux

d'interactions biologiques compte tenu de leur grande complexité.

Ma dernière contribution a été de tester et montrer l'applicabilité de l'inférence de modèles avec BoNesis. Dans le chapitre 5, je présente deux applications auxquelles j'ai participé, réalisées sur des données réelles qui ont permis d'extraire les interactions d'intérêt parmi un ensemble de connaissances afin de modéliser le comportement biologique observé. Des données sur des cellules cancéreuses, prévues au commencement de la thèse, n'ont finalement pas pu être obtenues ; leur production ayant été contrariée par des contraintes souvent liées à ce type de projets d'expérimentation. Cela n'a cependant eu d'impact ni sur la prise en compte de l'enjeu de la modélisation de processus cancéreux lors du développement de la méthode, ni sur l'applicabilité de la méthode à ce type de problématique. La première application aborde en effet directement ce second point avec une problématique de modélisation en cancérologie. Elle a été réalisée en reprenant les données utilisées pour un précédent modèle, sur le déclenchement de l'invasion et de la migration de cellules cancéreuses du poumon. L'objectif a été de montrer ce qu'apporte la prise en compte d'un ensemble de modèles lors de la réalisation de prédictions. Pour la seconde application, j'ai utilisé des mesures expérimentales collectées sur des cellules en différenciation. L'objectif a été de montrer la méthodologie permettant, grâce à BoNesis, de modéliser des transitions cellulaires complexes. L'applicabilité montrée ici ne se limite pas à la biologie du développement mais s'étend également tout particulièrement à des problématiques de recherche en cancérologie. En effet, non seulement les données prises en compte pour cette application sont de la même nature que celles initialement prévues sur les cellules cancéreuses, mais l'application cible également le processus de différenciation cellulaire. J'illustre ainsi l'inférence de modèles pouvant aider à analyser la régulation des décisions cellulaires, que cette différenciation soit saine ou qu'elle mène à une pathologie.

Chapitre 2

Formalisme de la modélisation : le réseau booléen

Sommaire

Un comportement biologique est le résultat de l'interaction de composants biologiques. Dans le but de comprendre de quelle façon les interactions s'organisent pour réguler un comportement, les réseaux booléens sont largement adoptés comme modèles dynamiques de réseaux de régulation. Ce formalisme correspond à la granularité des connaissances actuelles en biologie, mais il a également des propriétés dynamiques particulièrement adaptées pour représenter une grande diversité de comportements biologiques.

D'une part, un réseau booléen est adapté à l'échelle et à la précision des observations sur les interactions moléculaires. En effet, grâce aux différentes techniques d'observations, on recueille des informations sur des composants biologiques fonctionnels telles que les gènes, leurs transcrits ou encore les protéines. On observe également des ensembles de caractères identifiant le type des cellules observées ainsi que leur comportement (par exemple : en prolifération, en différenciation, en apoptose...). Enfin, peuvent s'ajouter des informations sur les conditions environnementales telles que d'éventuelles perturbations du milieu de vie des cellules ou des perturbations des cellules elles-mêmes (par exemple : milieu en anaérobie, ajout d'un médicament, dommage ADN...). Bien que ces informations soient majoritairement quantitatives, leur niveau de précision rend pertinent de les analyser au travers d'une interprétation biologique qui correspond à une indication de présence, d'expression ou d'activité du composant observé via une mesure de signal ou un comptage. Les observations sont de ce fait ramenées à des valeurs booléennes décrivant différents couples d'états selon la nature du composant et de l'observation : absent/présent, inhibé/exprimé ou encore inactif/actif.

D'autre part, bien que gros grain, un réseau booléen est un formalisme qui intéresse les modélisateurs pour ses propriétés dynamiques. Des comportements variés peuvent être interprétés par des propriétés au sein de la dynamique d'un réseau booléen, y compris des comportements complexes tels que ceux observés lors du suivi de l'évolution des états des composants biologiques au cours de la différenciation cellulaire.

Dans ce chapitre, je vais définir ce qu'est un réseau booléen, indiquer comment calculer la dynamique d'un réseau booléen et quelles sont les propriétés dynamiques que ce formalisme peut produire.

2.1 La structure du réseau

2.1.1 Formule booléenne

La condition d'activation d'un composant est exprimée sous la forme d'une formule booléenne construite à partir des connecteurs logiques *"et"* ($\wedge$), *"ou"* ($\vee$) et "non" ($\neg$) et de variables booléennes pouvant être évaluées à 1 ou 0. Ainsi, a, $\neg b$, $a \wedge \neg b$ ou encore $a \vee \neg b$ sont des exemples de formules booléennes. Si les valeurs de toutes les variables booléennes sont données, une unique valeur booléenne peut être déterminée pour la formule booléenne.

La formule booléenne peut être exprimée sous forme normale conjonctive *(CNF)* qui est une conjonction de clauses disjonctives, une clause disjonctive étant alors une disjonction de variables booléennes ou de leur négation, et la forme normale disjonctive *(DNF)* qui est une disjonction de clauses conjonctives, une clause conjonctive étant une conjonction de variables booléennes ou de leur négation. Toute formule écrite dans une forme normale peut être traduite dans l'autre forme normale. Ainsi, la formule $(a \vee \neg b) \wedge c$ est sous forme normale conjonctive et correspond à la formule $(a \wedge c) \vee (\neg b \wedge c)$ sous forme normale disjonctive. Une DNF (resp. CNF) est minimale s'il n'est pas possible de la simplifier en une DNF (resp. CNF) équivalente de taille plus petite.

Dans le contexte biologique, la formule booléenne associée à un composant représente les coopérations et compétitions entre composants pour l'influencer. Formellement, le réseau d'interactions biologiques est modélisé sous la forme d'un réseau booléen composé de fonctions booléennes locales. Chaque fonction locale f_i associée à un composant i est une formule booléenne qui indique la condition nécessaire et suffisante pour déterminer l'état de ce composant i en fonction de l'état d'autres composants du réseau.

2.1.2 Réseau booléen

Définition 2.1.1 (Réseau booléen *(RB)*)**.**
Un réseau booléen de dimension n est une fonction $f : \mathbb{B}^n \to \mathbb{B}^n$ *avec* $\mathbb{B} = \{0, 1\}$*. Cette fonction est composée de* n *fonctions booléennes locales* $f_i : \mathbb{B}^n \to \mathbb{B}$ *pour* $i \in \{1, \ldots, n\}$*.* f_i *est la fonction locale du* i^e *composant.*

$$
\begin{aligned}
f_1(x) &= \neg x_4 \\
f_2(x) &= \neg x_1 \wedge \neg x_3 \wedge x_4 \\
f_3(x) &= \neg x_2 \\
f_4(x) &= \neg x_1
\end{aligned}
$$

FIGURE 2.1 – Exemple de réseau booléen f de dimension 4

Exemple La figure 2.1 montre un exemple de réseau booléen de dimension 4, c'est-à-dire constitué de 4 fonctions locales, chacune associée à un composant du réseau.

La fonction locale associée au 1er composant est déterminée par la négation de la valeur associée au 4ème composant. Ainsi, $f_1(x)$ est déterminé à 1 si seulement si, en entrée, le vecteur de valeurs booléennes contient la valeur 0 en 4ème position quelles que soient les autres valeurs dans le vecteur (en symbolisant par $*$ une valeur non déterminée 0 ou 1, pour tout vecteur $x = {*}{*}{*}0$ on a $f_1(x) = 1$).

La présence du connecteur logique $\wedge$ (*"et"*) entre les variables booléennes de la seconde fonction locale implique qu'elles doivent toutes être à 1 pour que la condition soit respectée. Ainsi, $f_2(x)$ est déterminé à 1 si et seulement si à la fois les composants 1 et 3 sont à l'état 0 et le composant 4 à l'état 1 (0001 et 0101 sont donc les deux vecteurs pour lesquels $f_2(x) = 1$).

2.1.3 Configuration d'un réseau booléen

Une *configuration* au sein d'un réseau booléen est un vecteur associant à chaque composant du réseau booléen une valeur décrivant son état.

Les sémantiques classiques, introduites en sections 2.2.1 et 2.2.2, considèrent deux valeurs possibles d'états pour les composants : 0 et 1. Une configuration au sein d'un réseau booléen désigne de ce fait usuellement un vecteur de valeurs booléennes dont la taille est égale à la dimension du réseau booléen.

Définition 2.1.2 (Configuration)**.** *Étant donné un réseau booléen f de dimension n, un vecteur de valeurs $x \in \mathbb{B}^n$ est appelé une configuration de f.*

La sémantique Most Permissive, introduite en section 2.2.3, considère deux états dit dynamiques ($\nearrow$ et $\searrow$) en plus des états booléens. Je nomme de ce fait *configuration dynamique* un vecteur pouvant inclure ces quatre valeurs d'états.

Définition 2.1.3 (Configuration MP)**.** *Étant donné un réseau booléen f de dimension n, un vecteur de valeurs $X \in \mathbb{P}^n$ avec $\mathbb{P} = \{0, \nearrow, \searrow, 1\}$ est appelé une configuration MP de f.*

Notation L'ensemble des composants qui diffèrent entre deux configurations x et y est noté $\Delta(x, y) = \{i \in [1, n] \mid x_i \neq y_i\}$.

Exemple En considérant un réseau booléen f de dimension 4 tel que celui présenté en figure 2.1, les vecteurs 0000 et 0101 sont deux exemples de configurations et $\Delta(0000, 0101) = \{2, 4\}$. Le vecteur $0 \nearrow 01$ est un exemple de configuration MP.

On dit qu'une configuration x est associée à une configuration MP m si l'ensemble des composants ayant un état booléen (0 ou 1) dans m ont un état identique dans x.

Définition 2.1.4 (configuration associée à une configuration MP)**.** *Étant donné un réseau booléen f de dimension n, une configuration x est dite associée à une configuration MP $m = (m_1, \ldots, m_n)$ si et seulement si pour tout composant $i \in \{1, \ldots, n\}$, $m_i \in \{0, 1\} \Rightarrow x_i = m_i$. La fonction $\gamma : \mathbb{P}^n \to 2^{\mathbb{B}^n}$ associe à toute configuration MP $m \in \mathbb{P}^n$ l'ensemble des configurations qui lui sont associées : $\gamma(m) = \{x \in \mathbb{B}^n | \forall i \in \{1, \ldots, n\}, x_i \in \mathbb{B} \Rightarrow x_i = m_i\}$.*

Exemple 00 et 01 sont les configurations associées à la configuration MP $0 \nearrow$, donc $\gamma(0 \nearrow) = \{00, 01\}$. $\gamma(1 \nearrow 0) = 100, 110$ et $\gamma(\nearrow\searrow\nearrow) = \mathbb{B}^3$.

2.1.4 Graphe d'interactions d'un réseau booléen

Étant donné que la fonction locale de chaque composant dépend typiquement d'un petit sous-ensemble de composants, il est possible de représenter les dépendances entre composants au sein d'un réseau booléen par un

graphe d'interactions de ce réseau booléen tel que précédemment introduit en section 1.2.1.1 et illustré par la figure 2.2.

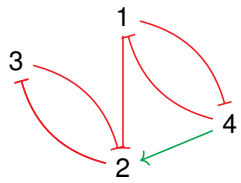


FIGURE 2.2 – Graphe d'interactions de f défini en fig.2.1

Les nœuds représentent les composants. Un arc positif (respectivement négatif) du nœud j vers le noeud i est défini s'il y a des configurations pour lesquelles la seule augmentation de la valeur du composant j augmente (resp. diminue) la valeur du composant i.

Définition 2.1.5 (Graphe d'interactions - *GI*)**.**
Étant donné un réseau booléen f de dimension n, son graphe d'interactions $G(f)$ est un graphe orienté $(\{1,\ldots,n\}, E_+, E_-)$ avec E_+ (resp. E_-) l'ensemble des arcs positifs (resp. négatifs) tels que $(j,i) \in E_+$ (resp. $(j,i) \in E_-$) si et seulement si $\exists x, y \in \mathbb{B}^n$ tels que $\Delta(x,y) = \{j\}, x_j < y_j$, et $f_i(x) < f_i(y)$ (resp. $f_i(x) > f_i(y)$).

Remarque Il est important de noter que des réseaux booléens différents peuvent avoir le même graphe d'interactions. Ainsi le réseau booléen h de dimension 4 qui ne diffère du réseau booléen f défini en figure 2.1 que par sa 2è fonction locale avec $h_2(x) = (\neg x_1 \wedge \neg x_3) \vee x_4$, partage le même graphe d'interactions que celui de f présenté en figure 2.2.

2.1.5 Réseau booléen localement monotone

Pour la création de modèles de réseaux d'interactions biologiques, nous nous intéressons uniquement aux réseaux booléens localement monotones. Dans un réseau booléen localement monotone si un composant i dépend d'un composant j, alors la fonction locale f_i (sous forme normale minimale) dépend toujours positivement ou toujours négativement de j. Au sein de la formule booléenne de la fonction locale, cela signifie qu'il ne peut pas y avoir deux variables booléennes représentant un même composant à la fois avec et sans négation. La figure 2.1 est un exemple de RB localement monotone avec $n = 4$: aucune variable n'apparaît deux fois au sein d'une formule booléenne, une fois associée à une négation et une fois sans négation. Ça signifie également qu'au sein du graphe d'interactions *(GI)* du réseau booléen introduit en section 2.1.4, étant donné deux sommets x et y, il ne peut pas y avoir deux arcs de signes différents de x vers y à l'image du GI représenté en figure 2.2.

Définition 2.1.6 (Réseau booléen localement monotone)**.**
Un réseau booléen $f = f_1, \ldots, f_n$ est dit localement monotone si et seulement si, étant donné $G(f) = (\{1,\ldots,n\}, E_+, E_-)$ son graphe d'interaction, $E_+ \cap E_- = \emptyset$.

En pratique la grande majorité des réseaux booléens modélisant des réseaux biologiques sont localement monotones car une hypothèse de modélisation est que la non-monotonie vient d'une compression du réseau, lorsqu'on ne fait pas apparaître certains complexes par exemple. Ainsi une fonction locale non monotone telle que $f_1(x) = (\neg x_2 \wedge x_3) \vee (x_2 \wedge \neg x_3)$ cache souvent un complexe x_{2+3} qui inhibe le composant 1 alors que les formes indépendantes des composants 2 et 3 activent le composant 1. Considérer uniquement les réseaux booléens localement monotones permet de ne pas augmenter la complexité des propriétés à aborder pour la modélisation et apparaît raisonnable étant donné le niveau de détail des réseaux d'interactions de gènes tels qu'étudiés pour la modélisation de différenciations cellulaires.

2.1.6 Mutation au sein d'un réseau booléen

Un réseau booléen peut être soumis à des perturbations permanentes de ses composants, appelées mutations. Ces mutations peuvent être un gain de fonction (GoF ; activité forcée du composant) ou une perte de fonction (LoF ; inactivité forcée du composant).

Définition 2.1.7 (Réseau booléen muté)**.**
Une mutation est un couple (i, v) où i est un composant du RB et $v \in \mathbb{B}$ est sa valeur forcée. On considère un RB f de dimension n et un ensemble de mutations M tel que pour tout $i \in \{1, \ldots, n\}$, il n'existe pas $\{(i, 0), (i, 1)\} \in M$. Nous notons par f/M le RB muté où, pour tout composant $i \in \{1, \ldots, n\}$ et $\forall x \in \mathbb{B}^n$, $(f/M)_i(x) = v$ si $(i, v) \in M$, et $(f/M)_i(x) = f_i(x)$ sinon.

Exemple L'observation d'un système biologique peut être associée aux informations suivantes : une mutation génétique inactive un gène 1 et une perturbation expérimentale pérennise l'activation d'un gène 2. On cherchera alors à représenter cette information par un réseau booléen muté avec $M = \{(1, 0), (2, 1)\}$.

2.2 Sémantiques

Un réseau booléen peut être considéré comme un système dynamique discret lorsqu'il est associé à une sémantique qui spécifie comment calculer l'évolution de l'état de ses composants et donc les transitions entre les configurations. En effet, étant donné une configuration, la sémantique décrit comment calculer la (ou les) prochaine(s) configuration(s) possible(s) en une *transition*, c'est-à-dire en un pas de temps du système. Pour cela, une sémantique définit combien de composants du réseau sont mis à jour à chaque transition, ; elle peut imposer l'ordre dans lequel ils sont mis à jour et décrire la façon dont un composant change d'état.

Les sémantiques classiques, appelées synchrone et asynchrones, décrivent les composants selon deux états booléens possibles. Elles diffèrent entre elles par le nombre de composants mis à jour simultanément. La sémantique Most Permissive introduite en [Paulevé et al., 2020] diffère d'une sémantique classique en décomposant le

changement d'état en deux temps, décrivant les composants selon quatre états possibles.

2.2.1 Sémantique synchrone

La sémantique la plus connue est la sémantique synchrone. À chaque pas de temps, la mise à jour simultanée de l'ensemble des composants constitue une transition.

Définition 2.2.1 (Transition synchrone)**.** *Étant donné un réseau booléen f de dimension n avec la sémantique synchrone s, $\forall x, y \in \mathbb{B}^n$, il existe une transition de x vers y, notée $x \xrightarrow[s]{f} y$, si et seulement si $x \neq y$ et $y = f(x)$.*

La sémantique synchrone est déterministe : il n'existe toujours qu'une seule configuration possible en un pas. Elle apparaît cependant peu réaliste pour modéliser des comportements biologiques qui impliquent par nature des interactions se déroulant à des vitesses différentes.

2.2.2 Sémantiques asynchrones

Les sémantiques asynchrones (générale et pleinement asynchrone), où seuls certains composants sont mis à jour à chaque pas de temps, sont communément considérées comme plus réalistes biologiquement car elles capturent différentes échelles temporelles dans les interactions. Ce sont des modes de mises à jour non déterministes où plusieurs configurations sont possibles selon les fonctions locales appliquées. On distingue deux sémantiques asynchrones.

La sémantique **asynchrone générale** considère n'importe quel sous-ensemble de composants pour la mise à jour à chaque pas de temps.

Définition 2.2.2 (Transition asynchrone générale)**.** *Étant donné un réseau booléen f de dimension n avec la sémantique asynchrone générale g, $\forall x, y \in \mathbb{B}^n$, $x \xrightarrow[g]{f} y$ si et seulement si $x \neq y$ et $\forall i \in \Delta(x, y), y_i = f_i(x)$.*

La sémantique **pleinement asynchrone** *(fully-asynchronous semantics)* considère l'application d'une seule fonction locale par pas de temps.

Définition 2.2.3 (Transition pleinement asynchrone)**.** *Étant donné un réseau booléen f de dimension n avec la sémantique pleinement asynchrone a, $\forall x, y \in \mathbb{B}^n$, $x \xrightarrow[a]{f} y$ si et seulement si $\exists i \in [1, n] : \Delta(x, y) = \{i\}$ et $y_i = f_i(x)$.*

Il existe de nombreuses variantes aux sémantiques classiques précisant un ordre dans l'application des fonctions locales ou des sous-ensembles de fonctions simultanées. L'ensemble de ces sémantiques considèrent des états booléens et les évolutions d'états qu'elles calculent sont incluses dans les évolutions calculées par la sémantique asynchrone générale. Pourtant, malgré sa grande utilisation pour la modélisation en biologie et bien qu'on pourrait penser qu'en considérant une à une toutes les mises à jour possibles cela suffit à aborder tous les comportements possibles, [Paulevé et al., 2020] a démontré que la sémantique asynchrone des réseaux booléens n'est pas une

abstraction fidèle pour les systèmes quantitatifs : elle passe à côté de comportements biologiquement pertinents en ne rendant pas compte de toutes les échelles temporelles et niveaux intermédiaires possibles. Pour pallier ce problème, [Paulevé et al., 2020] définit un nouveau mode de mise à jour appelé sémantique *Most Permissive*.

2.2.3 Sémantique *Most Permissive*

Il existe deux formulations équivalentes de la sémantique *Most Permissive* (MP). Je présente ci-dessous les deux formulations car elles apportent des notions complémentaires pour la compréhension de cette dynamique et pour les propriétés ensuite exploitées dans le cadre des travaux de thèse. La première formulation, la plus intuitive, introduit la notion d'états dynamiques. La seconde, qui clarifie que la sémantique MP n'est pas un réseau multivalué, introduit la définition d'hypercube.

2.2.3.1 Formulation avec états dynamiques

La sémantique *Most Permissive* (MP) se distingue de la sémantique pleinement asynchrone en considérant deux valeurs supplémentaires pour décrire l'état des composants : $\nearrow$ pour croissant et $\searrow$ pour décroissant, considérant de ce fait l'ensemble de valeurs $\mathbb{P} = \{0, \nearrow, \searrow, 1\}$. Un composant décrit par l'un de ces états dit *dynamiques* peut être interprété au choix comme étant à 0 ou 1. Ces états dynamiques sont le résultat d'un changement d'état des composants décomposé en deux temps.

Dans un premier temps, le composant dont la fonction locale détermine un changement de valeur passe par un état dynamique qui indique l'augmentation ($\nearrow$) ou la diminution ($\searrow$) de sa valeur. Un composant i de valeur 0 (resp. 1) obtiendra ainsi la valeur $\nearrow$ (resp. $\searrow$) si le calcul de sa fonction locale f_i donne 1 (resp. 0). Lorsqu'un composant est dans un état dynamique, les autres composants du réseau peuvent arbitrairement l'évaluer à la valeur 0 ou 1. Ce passage par une valeur ambiguë permet de représenter l'absence d'information sur les seuils à atteindre pour que l'interaction entre deux composants biologiques ait lieu. Par exemple, si un composant i est dans l'état $\nearrow$, un composant j peut l'évaluer à 1 alors qu'un composant k peut l'évaluer à 0. Ceci permet de considérer implicitement le fait que i peut avoir différents seuils d'activation selon les composants du réseau.

Ce n'est que dans un second temps qu'un composant retrouve une valeur booléenne : un composant qui est dans l'état $\nearrow$ (resp. $\searrow$) peut atteindre 1 (resp. 0).

Définition 2.2.4 (sémantique Most Permissive)**.** *Étant donné un réseau booléen f de dimension n avec la sémantique Most Permissive mp, $\forall x, y \in \mathbb{P}^n$, $x \xrightarrow[mp]{f} y$ si et seulement si $\exists i \in \{1, \ldots, n\} : \Delta(x, y) = \{i\}$ et*

$$y_i = \begin{cases} \nearrow & \textit{si } x_i \neq 1 \wedge \exists z \in \gamma(x) : f_i(z) = 1 \\ 1 & \textit{si } x_i = \nearrow \\ \searrow & \textit{si } x_i \neq 0 \wedge \exists z \in \gamma(x) : f_i(z) = 0 \\ 0 & \textit{si } x_i = \searrow \end{cases} \quad \textit{où } \gamma(x) \textit{ est défini en 2.1.4.}$$

2.2.3.2 Formulation avec hypercubes

L'ajout des états dynamiques $\nearrow$ et $\searrow$ pourrait suggérer que la sémantique MP est proche d'un réseau multivalué avec 4 valeurs. Mais contrairement aux réseaux multivalués, les états $\mathbb{P}$ ne sont pas totalement ordonnés par les transitions autorisées par la sémantique. Il existe une définition équivalente de la sémantique MP qui ne repose pas sur ces états dynamiques mais sur le calcul d'hypercubes clos par f. Un hypercube dans $\mathbb{B}^n$ a un ensemble de composants fixés à une valeur booléenne et les autres laissés libres (notés avec $*$).

Définition 2.2.5 (Hypercube)**.**
Un hypercube h de dimension n est un vecteur dans $(\mathbb{B} \cup \{*\})^n$. L'ensemble de ses configurations associées est noté $c(h) = \{x \in \mathbb{B}^n | \forall i \in \{1, \ldots, n\}, h_i \neq * \Rightarrow x_i = h_i\}$.

Étant donné deux hypercubes $h, h' \in (\mathbb{B} \cup \{*\})^n$, h est plus petit que h' si et seulement si $c(h) \subseteq c(h')$, ou de manière équivalente, $\forall i \in \{1, \ldots, n\}, h'_i \neq * \Rightarrow h_i = h'_i$. Un hypercube est minimal s'il n'existe pas d'hypercube différent plus petit que lui.

Exemple Le vecteur $h = 01*$ est un exemple d'hypercube de dimension 3, avec pour ensemble de configurations associées $c(h) = \{010, 011\}$. Étant donné $h' = 0**$, h est plus petit que h'.

On dit qu'un hypercube $h \in (\mathbb{B} \cup \{*\})^n$ est clos par $f : \mathbb{B}^n \to \mathbb{B}^n$ si pour toute configuration $x \in c(h), f(x) \in c(h)$. Un hypercube h clos par f, que nous appelons également *trap space*, est minimal si et seulement si c'est le plus petit hypercube clos par f contenant x pour tout $x \in c(h)$.

Exemple Considérons l'hypercube $01*$ ainsi que le réseau booléen $f : \mathbb{B}^3 \to \mathbb{B}^3$ suivant :

$$f_1(x) = \neg x_2$$

$$f_2(x) = \neg x_1$$

$$f_3(x) = \neg x_1 \wedge x_2$$

Tel que représenté sur la figure 2.3 :

- $01*$ est clos par f, avec $c(01*) = \{010, 011\}$. En effet, $f(010) = \{011\}$ et $f(011) = \{011\}$, donc pour toute configuration $x \in c(01*)$, $f(x) \in c(01*)$.
- $01*$ est le plus petit hypercube clos par f contenant 010.

- $01*$ n'est pas le plus petit hypercube clos par f contenant 011, puisque c'est 011 lui-même.

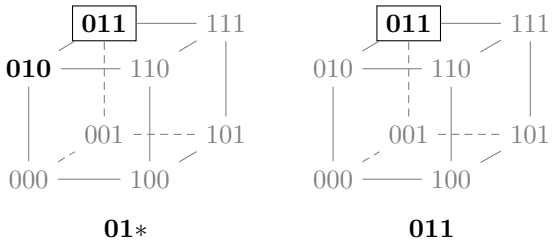


FIGURE 2.3 – Illustration des plus petits hypercubes contenant 010 (à gauche) et 011 (à droite) pour le réseau booléen f de dimension 3 avec $f_1(x) = \neg x_2$, $f_2(x) = \neg x_1$, $f_3(x) = \neg x_1 \wedge x_2$. Les configurations appartenant aux hypercubes sont indiquées en gras. Celles vérifiant la propriété d'atteignabilité en sémantique MP sont encadrées.

La définition d'hypercube clos est généralisée afin de n'imposer la clôture que pour un sous-ensemble de composants $K \subseteq \{1, \ldots, n\}$:

Définition 2.2.6 (Hypercube K-clos, aussi appelé *trap space* contraint sur K)**.**
Étant donné $K \subseteq \{1, \ldots, n\}$, un hypercube $h \in (\mathbb{B} \cup \{*\})^n$ est K-clos par f si pour toute configuration $x \in c(h)$, pour tout composant $i \in K$, $h_i \in \{*, f_i(x)\}$.

Remarque : Un hypercube h est clos par f si et seulement si h est $\{1, \ldots, n\}$-clos par f.

Exemple Considérons l'hypercube $h = 1*0$ et le réseau booléen f défini à l'exemple précédent. Tel que représenté sur la figure 2.4 :

- $1*0$ est le plus petit hypercube $\{2,3\}$-clos par f contenant 110. En effet, d'une part c'est bien un hypercube $\{2,3\}$-clos par f puisque, concernant $h_3 = 0$ et étant donné $c(1*0) = \{100, 110\}$, $f_3(100) = f_3(110) = 0$. Et d'autre part, il est bien le plus petit $\{2,3\}$-clos par f à contenir 110 puisque, concernant $h_2 = *$, $f_2(110) = 0$ donc 110 n'est pas un hypercube $\{2,3\}$-clos.
- $1*0$ n'est pas clos par f (c'est-à-dire $\{1,2,3\}$-clos) puisque $f_1(110) = 0$ et que $010 \notin c(1*0)$.
- $1*0$ n'est pas le plus petit hypercube $\{2,3\}$-clos par f contenant 100. En effet, 100 est lui-même un hypercube $\{2,3\}$-clos par f et $c(100) \subseteq c(1*0)$. Remarque : 100 est également clos par f (c'est-à-dire $\{1,2,3\}$-clos).

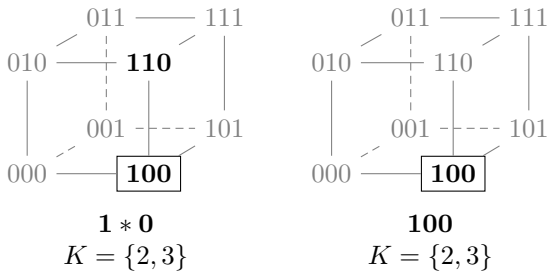


FIGURE 2.4 – Illustration des plus petits hypercubes $\{2,3\}$-clos contenant 110 (à gauche) et 100 (à droite) pour le réseau booléen f de dimension 3 avec $f_1(x) = \neg x_2$, $f_2(x) = \neg x_1$, $f_3(x) = \neg x_1 \wedge x_2$. Les configurations appartenant aux hypercubes sont indiquées en gras. Celles vérifiant la propriété d'atteignabilité en sémantique MP sont encadrées.

Partant d'une configuration x, la sémantique MP peut être définie via le calcul des plus petits hypercubes contenant x et qui sont K-clos par f, pour tout ensemble K, de la façon suivante :

- x est l'unique hypercube $\emptyset$-clos par f contenant x.
- Le changement d'état d'un composant $i \in \{1, \ldots, n\}$ vers $\nearrow$ ou $\searrow$ produit une configuration x' où $\gamma(x')$ correspond à l'hypercube $h \in (\mathbb{B} \cup \{*\})^n$ avec $h_i = *$ et, pour tout autre composant $j \in \{1, \ldots, n\}$ tel que $j \neq i$, $h_j = x_j$. Par conséquent, h est le plus petit hypercube $\{i\}$-clos par f et contenant x.
- En considérant uniquement les changements d'états vers $\nearrow$ ou $\searrow$, la sémantique agrandit progressivement l'hypercube selon les composants modifiés. Sont ainsi construits les plus petits hypercubes K-clos par f et contenant x, pour tout $K \subseteq \{1, \ldots, n\}$.

Avec la sémantique MP, le changement d'état pour un composant depuis un état dynamique ($\nearrow$ ou $\searrow$) vers un état booléen est sans condition et est seulement déterminé par l'état dynamique dans lequel il se trouve : 1 depuis $\nearrow$, 0 depuis $\searrow$. Par conséquent, en partant d'une configuration (booléenne), un composant peut être dans l'état $\nearrow$ seulement s'il y avait précédemment une configuration dynamique $x' \in \mathbb{P}^n$ telle qu'il existe une configuration $z \in \gamma(x')$ pour laquelle $f_i(z) = 1$ (resp. $\searrow$ si $f_i(z) = 0$).

Ainsi, avec la sémantique MP, une configuration $y \in \mathbb{B}^n$ est atteinte depuis $x \in \mathbb{B}^n$ si et seulement s'il existe un ensemble de composants $K \subseteq \{1, \ldots, n\}$ tel que le plus petit hypercube $h \in (\mathbb{B} \cup \{*\})^n$ K-clos par f et contenant x vérifie les deux conditions suivantes :

1. h contient y,
2. pour tout composant $i \in K$, il existe une configuration z dans h telle que $f_i(z) = y_i$.

2.2.3.3 Apports de la sémantique MP

Afin de pallier le fait que les réseaux booléens associés aux sémantiques classiques ne permettent pas de capturer tous les comportements du système quantitatif modélisé, plusieurs cadres de modélisation introduisent une granularité plus fine. Il existe par exemple les réseaux multivalués où les composants peuvent prendre plus que deux valeurs logiques, mais également les *fuzzy logic* qui étendent les modèles logiques au domaine continu ainsi que les équations différentielles ordinaires où les valeurs des composants sont des réels non négatifs qui varient le long d'un temps continu. Mais pour être spécifiés, ces systèmes requièrent des informations qui sont rarement connues lorsqu'il s'agit de modéliser un système biologique, telles que les seuils d'activation ou la cinétique des réactions. La sémantique *Most Permissive* étend les possibilités de modélisation sous la forme de réseau booléen, en conservant l'avantage de la nature gros-grain de ce formalisme sans requérir de formuler des hypothèses non étayées par les connaissances sur le système biologique modélisé.

Il est en effet démontré dans [Paulevé et al., 2020] qu'elle offre la garantie de pouvoir reproduire avec un réseau booléen tous les comportements réalisables par n'importe quel cadre de modélisation plus fin, tel que ceux cités

précédemment. Cela signifie que si un changement d'état est impossible avec la sémantique Most Permissive, il n'existe pas de modèle qualitatif ou quantitatif plus précis (issu d'un raffinement du RB) qui est capable de réaliser ce changement d'état. Il a également été montré que s'il existe un changement d'état possible, alors il existe un raffinement multivalué du réseau booléen qui reproduit l'équivalent multivalué de ce changement d'état au sein d'une trajectoire asynchrone.

La sémantique Most Permissive offre un autre avantage majeur levant une limite importante pour les applications en biologie des systèmes : sa moindre complexité pour décider l'existence de propriétés dynamiques élémentaires par rapport aux sémantiques classiques, comme nous le mentionnerons dans la section 2.3.5.

2.3 Propriétés dynamiques

Pour la suite du manuscrit, nous allons nous focaliser sur deux propriétés dynamiques principales des réseaux booléens : l'*atteignabilité* qui se rapporte à la possibilité pour une configuration d'évoluer en une autre configuration, et des comportements de *stabilité* qui se rapportent à une ou plusieurs configurations stables dans les comportements limites du système ou à un sous-ensemble de composants aux valeurs stables au sein d'un espace de la dynamique.

2.3.1 Atteignabilité

Définition 2.3.1 (configuration atteignable)**.** *Étant donné un réseau booléen f avec une sémantique σ et deux configurations ou configurations MP x et y, on dit que y est atteignable depuis x si et seulement si $x \xrightarrow[\sigma]{f}{}^{*} y$ où $\xrightarrow[\sigma]{f}{}^{*}$ est la fermeture transitive et réflexive de $\xrightarrow[\sigma]{f}$.*

Notation L'ensemble des configurations atteignables à partir d'une configuration x est noté $\rho^f_\sigma(x) = \{y \in \mathbb{B}^n \mid x \xrightarrow[\sigma]{f}{}^{*} y\}$.

Exemple Étant donné le réseau booléen f défini en figure 2.1, l'ensemble des configurations atteignables depuis 0000 sont $\rho^f_a(0000) = \{1000, 0000, 0010, 0001, 1010, 0011, 0101\}$. La configuration initiale 0000 fait partie de l'ensemble des configurations atteignables puisqu'on considère la fermeture transitive mais également réflexive de $\xrightarrow[\sigma]{f}$.

Les atteignabilités entre configurations d'un réseau booléen soumis à une sémantique donnée peuvent être représentées sous la forme d'un *graphe de transitions* dont les sommets sont les configurations et les arcs sont les transitions. Le graphe de transitions du réseau booléen f en figure 2.1 avec la sémantique asynchrone est représenté en figure 2.5.

Définition 2.3.2 (Graphe de transitions)**.** *Étant donné un réseau booléen f de dimension n avec une sémantique σ, le graphe de transitions a pour sommets toutes les configurations possibles de taille n et pour arcs toutes les transitions $\xrightarrow[\sigma]{f}$ entre configurations.*

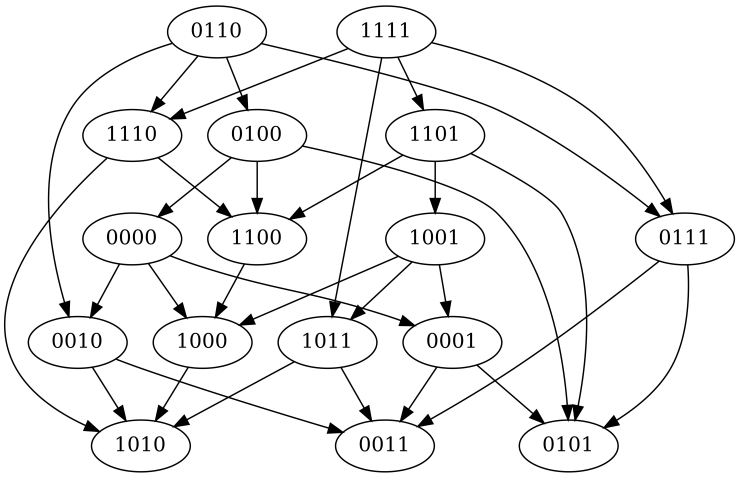


FIGURE 2.5 – Graphe de transition du réseau booléen f défini en fig.2.1 en sémantique pleinement asynchrone.

2.3.2 Trajectoire

Une trajectoire au sein de la dynamique d'un réseau booléen est une séquence finie de configurations telles qu'on passe de l'une à l'autre par l'application d'une seule transition.

Définition 2.3.3 (Trajectoire)**.** *Étant donné un réseau booléen f avec une sémantique σ, une séquence finie de configurations $\mathcal{T}_\sigma^f = (c_1, c_2, \ldots, c_l)$ est appelée trajectoire dans f selon σ si et seulement si pour tout $k \in [1, l-1]$, $c_k \xrightarrow[\sigma]{f} c_{k+1}$.*

Exemple Étant donné le RB f présenté en figure 2.1, la séquence de configurations 1101, 1001 puis 0001 est une trajectoire au sein de la dynamique du réseau booléen f défini en figure 2.1 avec la sémantique pleinement asynchrone, visible sur le graphe des transitions en figure 2.5.

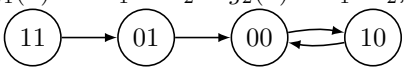
2.3.3 Attracteur

Les comportements limites d'un réseau booléen sont communément caractérisés par ce qu'on appelle les *attracteurs*. Ils correspondent aux plus petits ensembles de configurations clos par la relation d'atteignabilité.

Définition 2.3.4 (Attracteur : point fixe et attracteur cyclique)**.** *Étant donné un réseau booléen f de dimension n avec une sémantique σ, un ensemble A non-vide de configurations est un attracteur de f avec σ si et seulement si $\forall x \in A, \rho_\sigma^f(x) = A$. Un attracteur constitué d'une seule configuration est appelé un point fixe du réseau booléen ($f(x) = x$) ; il est sinon qualifié d'attracteur cyclique.*

Remarque Au regard du graphe de transitions, les attracteurs correspondent aux composantes fortement connexes terminales du graphe ; points fixes si elles ne contiennent qu'un seul sommet, attracteurs cycliques sinon.

Exemple de point fixe Avec le réseau booléen f en sémantique pleinement asynchrone dont le graphe de transitions est présenté en figure 2.5, la configuration 1010 est un attracteur de type point fixe. En effet, quelle que soit la fonction locale considérée, $f(1010) = 1010$. Le sont également les configurations 0011 et 0101.

Exemple d'attracteur cyclique Dans le réseau booléen g de dimension 2 avec $g_1(x) = \neg x_1 \wedge \neg x_2$ et $g_2(x) = x_1 \wedge x_2$, le graphe de transitions en sémantique pleinement asynchrone est le suivant : 

Les configurations 00 et 10 constituent un attracteur cyclique puisque $\rho^f_{fa}(00) = \rho^f_{fa}(10) = \{00, 10\}$.

Il est important de noter que si une sémantique est non déterministe, il est possible d'atteindre plusieurs attracteurs à partir d'une configuration. C'est le cas par exemple au sein de la dynamique du réseau booléen f dont le graphe des transitions est présenté en figure 2.5 : à partir de la configuration 1111 il est possible d'atteindre les 3 points fixes 1010, 0011 et 0101. C'est grâce à cette caractéristique importante des réseaux booléens que ces derniers peuvent être utilisés pour la modélisation des processus de différenciation cellulaire.

Notation Étant donné un attracteur A dans un réseau booléen f avec une sémantique σ :

- $x \xrightarrow[\sigma]{f}{}^* A$ désigne l'existence d'un chemin de la configuration x vers une configuration appartenant à A ;
- $x \not\xrightarrow[\sigma]{f}{}^* A$ désigne l'absence de chemin de la configuration x vers une configuration appartenant à A.

2.3.4 Configuration confinée

La propriété de confinement au sein de la dynamique d'un réseau booléen permet de décrire l'impossibilité pour un ou plusieurs composants de changer de valeur à partir d'une configuration.

Définition 2.3.5 (configuration confinée sur un ensemble de composants)**.** *On considère un réseau booléen f de dimension n avec une sémantique σ, une configuration x et un ensemble de composants $I \subseteq \{1, \ldots, n\}$. Une configuration x de f est confinée sur I si pour toute configuration atteignable $x' \in \rho^f_\sigma(x)$ et $\forall i \in I$, $x'_i = x_i$.*

Remarques : Une configuration confinée n'appartient pas forcément à un attracteur puisque la dynamique peut encore se dérouler, mais il garantit que l'ensemble des attracteurs atteignables ont les composants I à la même valeur. Si x est confinée sur I, toute configuration $x' \in \rho^f_\sigma(x)$ est aussi confinée sur I.

Exemple Dans le réseau booléen f avec la sémantique pleinement asynchrone dont le graphe de transitions est présenté en figure 2.2, la configuration $x = 1110$ est confinée sur l'ensemble de composants $I = \{1, 4\}$ puisque toute configuration atteignable x' appartenant à $\rho^f_a(1110) = \{1100, 1000, 1010\}$ a $x'_1 = 1 = x_1$ et $x'_4 = 0 = x_4$. L'unique attracteur accessible depuis x est 1010.Toute configuration appartenant à $\rho^f_a(1110)$ est elle-même confinée sur $\{1, 4\}$.

2.3.5 Complexité

L'analyse de la dynamique des réseaux booléens repose le plus souvent sur trois propriétés fondamentales : les points fixes, caractérisant les configurations qui ne peuvent plus évoluer ; l'atteignabilité, caractérisant l'existence de trajectoires entre deux configurations données ; et les attracteurs, généralisant les points fixes et caractérisant les comportements limites du réseau. La complexité de ces propriétés pour la sémantique Most Permissive a été démontrée dans [Paulevé et al., 2020].

En sémantiques synchrone, pleinement asynchrone et asynchrone, le problème d'existence de point fixe est NP-complet, et décider si $y \in \rho^f_\sigma(x)$ ou si x appartient à un attracteur sont tous deux des problèmes PSPACE-complets [Paulevé et al., 2020]. En pratique, le calcul exact des attracteurs des réseaux booléens avec une sémantique classique passe difficilement à l'échelle. Selon la structure du réseau et la sémantique choisie, il est limité à 50 ou 100 composants. Avec la sémantique Most Permissive, la complexité de l'analyse d'atteignabilité et d'appartenance est considérablement réduite : décider si $y \in \rho^f_{mp}(x)$ est en temps polynomial si f est localement monotone et P^{NP} sinon. Décider s'il existe un point fixe demeure NP-complet mais la décision d'appartenance à un attracteur est réduit au plus à coNP pour les réseaux localement monotones et est dans $\mathrm{coNP}^{\mathrm{coNP}}$ dans le cas contraire [Paulevé et al., 2020]. En pratique, cela rend possible l'analyse formelle de réseaux de dimension de plusieurs ordres de grandeur supérieurs aux sémantiques classiques, analyse ainsi applicable pour la modélisation de processus biologiques où plusieurs milliers de composants sont considérés. Il est intéressant de relever les relations suivantes entre sémantiques MP et pleinement asynchrone :

— $x \in \mathbb{B}^n$ est un point fixe avec la sémantique MP si et seulement si c'est un point fixe avec la sémantique pleinement asynchrone ;

— $y \in \mathbb{B}^n$ est atteignable depuis $x \in \mathbb{B}^n$ avec la sémantique pleinement asynchrone seulement s'il est atteignable avec la sémantique MP ;

— le nombre d'attracteurs avec la sémantique MP est inférieur ou égal au nombre d'attracteurs avec la sémantique pleinement asynchrone.

2.4 Exemple récapitulatif

La dynamique des réseaux booléens est utilisée pour modéliser de façon gros grains la dynamique de systèmes biologiques, sans requérir de nombreux paramètres quantitatifs hors de portée des observations. Pour cette modélisation, les sémantiques synchrone et asynchrone sont fréquemment utilisées mais elles ne constituent pourtant pas une abstraction correcte des dynamiques quantitatives observées : elles amènent à prévoir des transitions parasites tout en excluant des transitions qui sont en fait possibles si l'on tient compte que les seuils d'activation peuvent différer d'un composant à l'autre en biologie. C'est en cela que la prise en compte de la

sémantique Most Permissive est importante dans la méthode d'inférence développée. Je vais illustrer les principales définitions présentées dans ce chapitre ainsi que les différences de dynamique selon la sémantique considérée en me basant sur le réseau booléen suivant :

$$f_1(x) = 1$$

$$f_2(x) = x_1$$

$$f_3(x) = (\neg x_1 \wedge x_2) \vee x_3$$

Le graphe d'interactions de f est montré en figure 2.6. Notons que ce réseau booléen f est localement monotone : aucun composant n'est en même temps activateur et inhibiteur d'un autre composant.

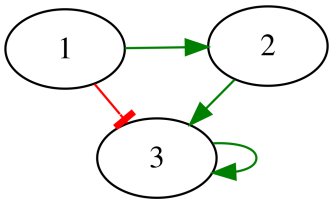


FIGURE 2.6 – Graphe d'interactions de f. Arc vert : activateur, arc rouge : inhibiteur.

Remarques sur ses propriétés dynamiques :

- Le réseau booléen possède deux points fixes qui sont les configurations 110 et 111.
- En considérant **la sémantique asynchrone**, il existe une trajectoire entre 000 et 110 ($000 \underset{a}{\overset{f}{\rightarrow}}{}^{*}\ 110$). En effet, à partir de 000, la seule transition possible est l'activation du premier composant, suivie à nouveau d'une unique transition possible qui est l'activation du second composant : $000 \underset{a}{\overset{f}{\rightarrow}}\ 100 \underset{a}{\overset{f}{\rightarrow}}\ 110$. Cette dernière configuration ne permet plus aucune transition. De ce fait, à partir de 000, l'activation du troisième composant est impossible ($000 \underset{a}{\overset{f}{\nrightarrow}}{}^{*}\ 111$) et 110 est le seul attracteur atteignable.
- Par contre, en considérant **la sémantique Most Permissive**, il existe bien une trajectoire entre la configuration 000 et l'attracteur 111 ($000 \underset{mp}{\overset{f}{\rightarrow}}{}^{*}\ 111$). Pour rappel, l'idée principale derrière la sémantique MP est de considérer un état intermédiaire, dit *dynamique*, lors de l'activation ($\nearrow$) ou la désactivation ($\searrow$) d'un composant. Ces états dynamiques peuvent être interprétés à 0 ou à 1. De ce fait dans l'exemple, lors de l'activation du premier composant à partir de la configuration 000, on atteint l'hypercube $\nearrow 00$. Le second composant peut alors être activé puisque $\nearrow$ peut être interprété à 1, menant à $\nearrow\nearrow 0$. Puis, le troisième peut être activé ($\nearrow\nearrow\nearrow$) : le premier composant peut en effet être interprété à 0 et le second à 1. Cette trajectoire modélise la situation biologique où le premier composant n'est pas assez exprimé pour inhiber le troisième, mais suffisamment pour activer le second. À partir de $\nearrow\nearrow\nearrow$, l'ensemble des composants peuvent basculer à 1, et comme 111 est un point fixe, l'analyse en sémantique MP conclut que deux attracteurs sont atteignables depuis 000 : $\{110\}$ et $\{111\}$.

Résumé du chapitre 2

Le réseau booléen constitue un cadre de modélisation adapté à la problématique de nombreuses applications en biologie. Ce formalisme est non seulement adapté à la précision des observations, mais de nombreux et divers comportements biologiques peuvent être interprétés par des propriétés au sein de la dynamique d'un réseau booléen. Dans ce chapitre, je donne la définition formelle d'un réseau booléen et je rappelle ses propriétés dynamiques élémentaires.

Un réseau booléen associe à chaque composant une fonction locale booléenne, qui prend en entrée un vecteur booléen appelé *configuration* associant à chacun de ses composants un état booléen. Les influences entre les composants d'un réseau booléen peuvent être résumées par un graphe d'interactions, où chaque arc (positif ou négatif) indique l'influence (activatrice ou inhibitrice) d'un composant sur un autre. Le graphe d'interactions correspondant à un réseau booléen ne fournit aucune information sur les connecteurs logiques composant la fonction associée à chaque composant, c'est pourquoi des réseaux booléens différents peuvent correspondre à un même graphe d'interactions. Je définis un réseau booléen muté comme un réseau booléen où la valeur de certains composants est forcée. Dans la suite de ce manuscrit, je considère seulement les réseaux booléens localement monotones, c'est-à-dire ceux pour lesquels chaque fonction locale ne dépend jamais à la fois positivement et négativement d'un même composant. La sémantique d'un réseau booléen permet de calculer les transitions possibles entre ses configurations, et ainsi l'évolution de l'état de ses composants au cours du temps. Je présente dans ce chapitre différentes sémantiques. Sous la sémantique synchrone, la plus connue, une transition implique l'application simultanée des fonctions locales de l'ensemble des composants du réseau booléen. Sous la sémantique asynchrone générale, n'importe quel sous-ensemble de composants est considéré, sous-ensemble composé d'un unique composant sous la sémantique pleinement asynchrone. La sémantique *Most Permissive* (MP) repose quant à elle sur la notion d'hypercube. Un hypercube est un sous-espace où un ensemble de composants sont fixés à une certaine valeur et les autres composants sont libres. Je le définis dans le chapitre sous la forme d'un vecteur, à l'image d'une configuration, à la différence qu'il peut contenir, en plus des valeurs booléennes, une valeur "libre" $*$.

Au sein de la dynamique des réseaux booléens, la problématique de modélisation abordée amène à se concentrer sur deux propriétés dynamiques principales : d'une part l'atteignabilité (c'est-à-dire l'existence de trajectoires) entre configurations et, d'autre part, l'existence d'attracteurs qui représentent les ensembles limites de configurations. Dans mon travail de thèse j'ai particulièrement considéré deux types de propriétés liées aux attracteurs : la propriété de point fixe qui caractérise des configurations d'où aucune transition n'est possible, et la propriété de confinement qui permet de capturer les attracteurs dans lesquels un sous-ensemble de composants ne peut pas osciller.

La méthode d'inférence de modèles présentée dans ce manuscrit considère la sémantique MP pour calculer la dynamique des réseaux booléens inférés automatiquement. En effet, la complexité de l'analyse de propriétés d'atteignabilité et d'attracteurs sous la sémantique MP rend abordable la modélisation de processus biologiques de plusieurs milliers de composants.

Chapitre 3

Cadre de modélisation de la différenciation cellulaire

Sommaire

La différenciation cellulaire, processus présenté en section 1.1.2 qui permet l'apparition de types différents de cellules, est caractérisée biologiquement par un ensemble de propriétés dynamiques. La différenciation englobe à la fois le fait qu'une cellule change de type en accumulant progressivement des différences par rapport à son état de départ, mais également le fait qu'une cellule mère se partage en deux cellules filles différentes l'une de l'autre. La différenciation cellulaire est indispensable pour la formation d'un organisme complexe, pour accompagner sa croissance comme pour le renouvellement de tissus tout au long de la vie de l'organisme. Cependant, ce processus

indispensable peut dévier et le risque est alors d'aboutir à des cellules dont les fonctions nuisent au bon fonctionnement de l'organisme. On peut observer la modification d'un tissu (celui de l'épithélium bronchique par exemple sous l'effet de traumatismes répétés tels que le tabac ou des infections) ou l'apparition de cellules cancéreuses qui accumulent les anomalies et échappent aux voies de régulation normales pour envahir progressivement l'organisme.

Il y a donc, au sein de la différenciation cellulaire, à la fois l'évolution graduelle d'un état, l'existence d'évolutions différentes à partir d'un même état initial, et la possibilité d'atteindre des états stables où les cellules cessent de maturer. Afin de pouvoir synthétiser des réseaux booléens modèles de ces processus biologiques complexes, je décris dans la première section de ce chapitre les propriétés caractéristiques des observations biologiques recueillies sur ces processus et je définis la compatibilité d'un réseau booléen avec chacune de ces propriétés. L'objectif étant d'être capable de modéliser des différenciations cellulaires, j'ai distingué trois propriétés majeures qui, combinées, permettent de caractériser ces processus.

La définition de ce cadre formel permet de raisonner sur le besoin auquel doit répondre une méthode d'inférence de réseaux booléens modélisant ces comportements complexes. Je termine le chapitre en présentant l'état de l'art des méthodes d'inférence de réseaux booléens à partir de données biologiques, afin d'exposer quels sont les types de propriétés dynamiques qu'elles abordent et ainsi quelle est la nature des comportements biologiques qui peuvent être modélisés. Au regard de ces méthodes, je souligne l'apport de celle que je développe dans la thèse.

3.1 Compatibilité entre réseau booléen et comportement biologique

Quelle que soit la technique de collecte des données, une observation d'un système biologique est un ensemble d'informations en lien avec la situation observée. Une situation peut correspondre à une condition expérimentale, un point de temps au sein d'un suivi temporel, ou encore à une cellule au sein d'une population cellulaire en différenciation. Concrètement, une observation d'un système biologique en différenciation peut contenir des états d'expression de gènes, le type de la cellule, ainsi que de potentielles perturbations permanentes telles que des mutations, ou temporaires telles que l'administration de molécules chimiques. Bien que peu de données issues des expérimentations soient booléennes par nature, une étape d'interprétation des données quantitatives telles que les mesures d'expression de gènes, aidée par des outils dédiés de binarisation, permet de déterminer l'état d'expression ou d'inhibition des gènes au sein des observations. Après cette étape d'interprétation des valeurs non booléennes, l'ensemble des informations peut être facilement présenté sous la forme d'un ensemble de couples associant à un composant une valeur booléenne. Cet ensemble de couples constitue une observation où des composants d'un système biologique sont associés à une valeur booléenne reflétant l'état observé de ces composants. À chaque composant ne peut être associée qu'une seule valeur, ainsi il est impossible qu'une observation indique à la fois l'expression et la non-expression d'un gène.

Définition 3.1.1 (Observation)**.** *On considère un système à n composants. Une observation o de k composants avec $1 \leq k \leq n$ est un ensemble de k couples (i, v), avec i l'identifiant du composant observé et $v \in \mathbb{B}$ sa valeur observée, tel qu'il n'existe pas de composant i tel que $\{(i, 0), (i, 1)\} \subseteq o$.*

3.1.1 Évolution cellulaire : liste d'observations

3.1.1.1 Information d'ordres entre les observations

Pour suivre l'évolution d'un processus biologique, les observations sont ordonnées les unes par rapport aux autres. Elles constituent ainsi une liste d'observations, soit parce qu'elles sont réalisées à différents points de temps d'une expérience, soit parce qu'un pseudo-temps a été estimé à partir d'observations à l'échelle de la cellule. La figure 3.1 représente une liste de quatre observations, après binarisation, permettant de suivre l'évolution des états de plusieurs composants d'un système. Pour rappel, une observation est un ensemble de couples $\{i, v\}$ associant un composant observé i à une valeur v tel que défini en 3.1.1. Au sein d'une liste, ce ne sont pas forcément les mêmes composants qui sont observés dans les différentes observations successives et le nombre de composants observés peut être différents.

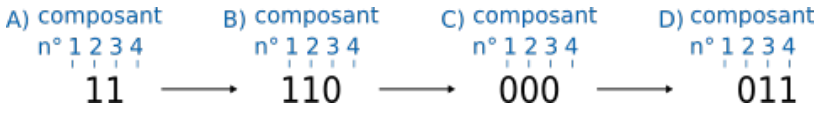


FIGURE 3.1 – Exemple d'une liste d'observations binarisées

3.1.1.2 Compatibilité avec un réseau booléen

Pour toute cette partie, on considère un réseau booléen de taille fixée n et des observations portant sur un nombre de composants compris entre 1 et n.

L'information apportée en figure 3.1 sur des composants du système biologique ne présume pas de l'état des composants non observés. De ce fait, une configuration est jugée compatible avec une observation expérimentale si l'ensemble des couples composants-valeurs sont retrouvés au sein de la configuration.

Définition 3.1.2 (Configuration compatible avec une observation)**.** *Étant donné un réseau booléen de dimension n, une configuration $x \in \mathbb{B}^n$ est dite compatible avec une observation o si et seulement si $\forall (i, v) \in o$, $x_i = v$.*

Par exemple, en considérant les observations présentées en figure 3.1, les configurations 1100 et 1101 sont compatibles avec l'observation B mais ne sont pas compatibles avec l'observation A puisque la valeur attribuée au composant 3 diffère dans ce cas avec l'observation. La figure 3.3 illustre la compatibilité entre configurations et observations en représentant le graphe de transition du réseau booléen de la figure 2.1 en sémantique pleinement asynchrone. Y sont mises en évidence par une même couleur les configurations compatibles avec les observations

présentées en figure 3.2.

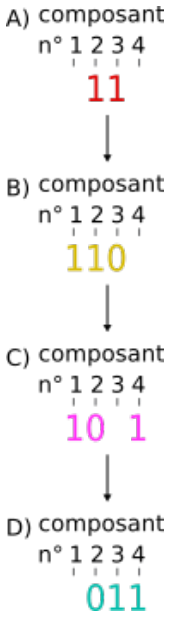


FIGURE 3.2 – Exemple d'une liste d'observations

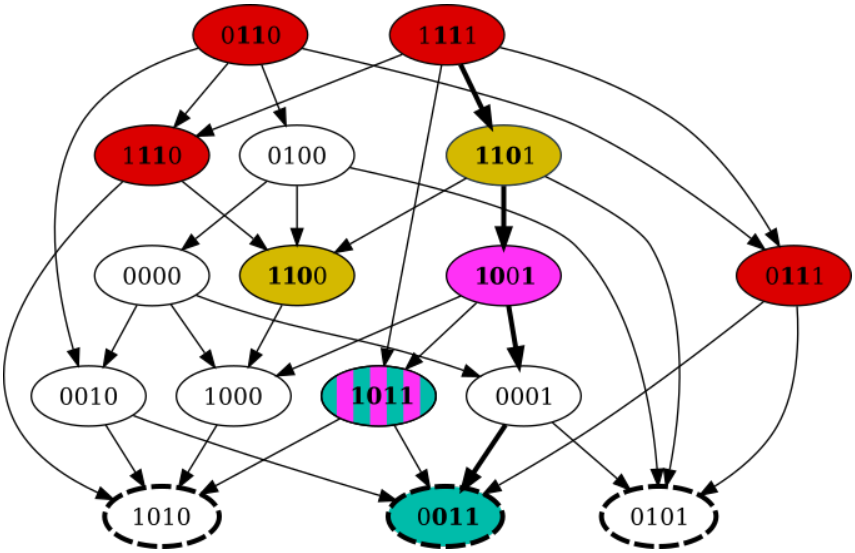


FIGURE 3.3 – Graphe de transition du RB f défini en fig.2.1 en sémantique pleinement asynchrone, avec coloration des configurations compatibles avec les observations de même couleur présentées en fig.3.2 et mise en gras des composants observés.

Trajectoire Afin de pouvoir définir la compatibilité d'un réseau booléen avec une liste d'observations, je définis la notion de trajectoire compatible avec une liste d'observations au sein d'un réseau booléen. Une trajectoire est compatible avec une liste d'observations si, en respectant l'ordre de la liste, on peut trouver des configurations compatibles avec les observations dans la trajectoire.

Définition 3.1.3 (Trajectoire compatible avec une liste d'observations)**.** *Étant donné un réseau booléen f avec une sémantique σ et une liste d'observations $\mathcal{S} = (o_1, \ldots, o_m)$, une trajectoire $\mathcal{T}^f_\sigma = (x_1, \ldots, x_l)$ est compatible avec $\mathcal{S}$ si et seulement s'il existe $i_1, \ldots, i_m \in \{1, \ldots, l\}$ avec $i_1 \leq i_2 \leq \ldots \leq i_m$ tel que pour tout $k \in \{1, \ldots, m\}$, la configuration x_{i_k} est compatible avec l'observation o_k.*

Remarque : au sein d'une trajectoire compatible avec une liste d'observations, une configuration peut être compatible avec plusieurs observations successives et plusieurs configurations peuvent être compatibles avec une même observation.

Exemple Le réseau booléen pleinement asynchrone dont le graphe de transitions est présenté en figure 3.3 produit une trajectoire $\mathcal{T}^f_a = (1111, 1101, 1001, 1011, 0011)$ compatible avec la liste d'observations $\mathcal{S} = (A, B, C, D)$ présentée en figure 3.2. En effet il s'y trouve au moins une configuration compatible avec chaque observation, dans un ordre qui respecte celui de la liste, par exemple : ${\mathcal{T}^f_a}_1$ est compatible avec $\mathcal{S}_1$, ${\mathcal{T}^f_a}_2$ est compatible avec $\mathcal{S}_2$, ${\mathcal{T}^f_a}_4$ est compatible avec $\mathcal{S}_3$ et ${\mathcal{T}^f_a}_5$ est compatible avec $\mathcal{S}_4$.

Interprétation de la représentativité de l'information Il est important de noter que la compatibilité d'un réseau booléen avec une liste d'observations est soumise à l'interprétation qui est faite de la représentativité des données par rapport au système observé, c'est-à-dire de la mesure dont les données collectées sont représentatives de l'ensemble des comportements possibles au sein du système biologique observé. La figure 3.2 illustre ainsi le besoin d'interpréter la portée de la propriété d'atteignabilité entre observations d'une même liste puisque plusieurs configurations sont compatibles avec une même observation et que plusieurs trajectoires peuvent être possibles à partir d'une même configuration. Or la seule liste d'observations n'indique pas si l'état du système qui a permis le comportement observé exclut ou non d'autres comportements possibles. Il existe de ce fait différents besoins d'interprétation que je décris par la flexibilité de considérer la compatibilité avec ou sans une notion d'*universalité* des propriétés recherchées au sein de la dynamique du réseau booléen.

Dans sa définition reposant sur l'hypothèse de représentativité la plus faible, un réseau booléen est dit compatible avec une liste d'observations s'il comprend une trajectoire compatible avec la liste d'observations.

Définition 3.1.4 (réseau booléen compatible avec une liste d'observations)**.** *Un réseau booléen f selon la sémantique σ est dit compatible avec une liste d'observations $\mathcal{S} = (o_1, o_2, \ldots, o_m)$ si et seulement s'il existe une configuration c_1 compatible avec o_1 et une trajectoire $\mathcal{T}_\sigma^f$ commençant par c_1 telle que $\mathcal{T}_\sigma^f$ est compatible avec $\mathcal{S}$.*

Exemple Le réseau booléen pleinement asynchrone dont le graphe de transitions est présenté en figure 3.3 est compatible avec la liste d'observations de la figure 3.2. En effet, depuis la configuration 1111 compatible avec l'observation A, $\mathcal{T}_a^f = (1111, 1101, 1001, 1011, 0011)$ est un exemple de trajectoire compatible avec la liste d'observations. On aurait pu aussi choisir $\mathcal{T}_a^f = (1111, 1101, 1001, 1011)$ qui est la plus petite trajectoire compatible possible, ou $\mathcal{T}_a^f = (1111, 1101, 1001, 0001, 0011)$.

L'hypothèse forte de la représentativité est décrite par l'ajout d'universalité sur la compatibilité. Un réseau booléen est dit *universellement* compatible avec une liste d'observations s'il existe une configuration compatible avec l'état initial à partir de laquelle toutes les trajectoires sont compatibles avec la liste d'observations ou sont le préfixe d'une trajectoire compatible.

Définition 3.1.5 (réseau booléen compatible universellement avec une liste d'observations)**.** *Un réseau booléen f selon la sémantique σ est dit compatible universellement avec une liste d'observations $\mathcal{S} = (o_1, o_2, \ldots, o_m)$ si et seulement s'il existe une configuration c_1 compatible avec o_1 telle que pour toute trajectoire $\mathcal{T}_\sigma^f$ commençant par c_1, soit $\mathcal{T}_\sigma^f$ est compatible avec $\mathcal{S}$, soit il existe une trajectoire $\mathcal{T'}_\sigma^f$ qui prolonge $\mathcal{T}_\sigma^f$ qui est compatible avec $\mathcal{S}$.*

Exemple Le réseau booléen pleinement asynchrone dont le graphe de transitions est présenté en figure 3.3 n'est pas universellement compatible avec la liste d'observations de la figure 3.2. En effet, quelle que soit la configuration c_1 compatible avec o_1, il existe une trajectoire qui n'est ni compatible avec la liste d'observations ni préfixe d'une trajectoire compatible :

- depuis 1111 : $1111 \rightarrow 1110 \rightarrow 1100 \rightarrow 1000$
- depuis 0111 : $0111 \rightarrow 0011$
- depuis 0110 : $0110 \rightarrow 0100 \rightarrow 0000$
- depuis 1110 : $1110 \rightarrow 1010$

3.1.2 Divergence d'évolution : bifurcation

3.1.2.1 Information de bifurcation entre listes d'observations

La différenciation cellulaire implique des phénomènes de bifurcation, c'est-à-dire des événements au cours desquels une cellule mère se partage en deux cellules filles différentes l'une de l'autre. Leur différence est irréversible, la maturation des cellules filles ne peut pas aboutir aux mêmes spécialisations et elles ne peuvent pas revenir à l'état de la cellule mère. Lorsqu'on observe un processus de différenciation cellulaire, les bifurcations impliquent que l'ensemble des observations peut être représenté sous une forme arborescente. La figure 3.4 illustre ainsi le suivi d'une différenciation cellulaire avec 6 observations correspondant à différentes étapes de la différenciation. Y sont observés les états de plusieurs composants qui ne sont pas nécessairement les mêmes entre observations.

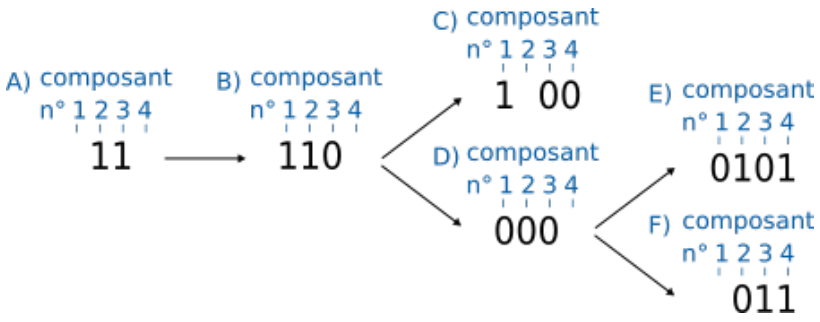


FIGURE 3.4 – Graphe représentant l'évolution de données de différenciation. L'état exprimé (1) ou inhibé (0) de plusieurs gènes est observé à différentes étapes de la différenciation cellulaire. La racine de l'arbre correspond à une cellule non différenciée, les feuilles correspondent aux cellules différenciées.

Une bifurcation implique un état initial que nous appelons *point de bifurcation*, à partir duquel il est possible d'observer différentes évolutions que nous appelons *voies de différenciation*.

Définition 3.1.6 (point de bifurcation et voie de différenciation)**.** *On considère un ensemble O de listes d'observations. Toute observation p pour laquelle il existe dans O deux listes d'observations s_1 et s_2 ayant pour suffixes $(p, p_1, \ldots, p_k)$ et $(p, p'_1, \ldots, p'_l)$ tels que $\{p, p_1, \ldots, p_k\} \cap \{p, p'_1, \ldots, p'_l\} = \{p\}$, est appelée point de bifurcation de s_1 et s_2, et chacun des suffixes $p_1, \ldots, p_k$ et $p'_1, \ldots, p'_l$ est appelé voie de différenciation issue de p.*

Exemple Sur la figure 3.4, on considère trois listes d'observations $s_1 = (A, B, C)$, $s_2 = (A, B, D, E)$ et $s_3 = (A, B, D, F)$. Pour s_1, s_2 le point de bifurcation est l'observation B et les voies de différenciation sont (C) et (D, E). Pour s_2, s_3 le point de bifurcation est l'observation D et les voies de différenciations sont (E) et (F).

Définition 3.1.7 (bifurcation associée à deux listes d'observations)**.** *Un ensemble de trois observations* $\{p, o_1, o_2\}$ *est appelé une bifurcation associée à deux listes d'observations* s_1, s_2 *si et seulement si* p *est un point de bifurcation de* s_1 *et* s_2 *dont* v_1, v_2 *sont les deux voies de différenciation issues de* p*, et* o_1, o_2 *sont deux observations appartenant respectivement à* v_1, v_2*.*

Exemple Sur la figure 3.4, $\{B, C, E\}$ est une bifurcation associée aux listes d'observations $s_1 = (A, B, C)$ et $s_2 = (A, B, D, E)$, avec B point de bifurcation de s_1, s_2 et avec l'observation E appartenant à la voie de différenciation $v_1 = (D, E)$ et l'observation C appartenant à la voie de différenciation $v_2 = C$. $\{B, E, F\}$ n'est pas une bifurcation associée aux listes d'observations $s_2 = (A, B, D, E)$ et $s_3 = (A, B, D, F)$ puisque B n'est pas un point de bifurcation pour s_1, s_2.

3.1.2.2 Compatibilité avec un réseau booléen

À l'instar des listes d'observations, les seules données de bifurcation n'indiquent pas si l'état du système qui a permis le comportement observé exclut ou non d'autres comportements possibles. Il existe de ce fait différents besoins d'interprétation que je décris par la flexibilité de considérer la compatibilité avec ou sans une notion d'*universalité* de la propriété au sein de la dynamique du réseau booléen. Dans l'hypothèse la plus faible, chaque voie de différenciation mène vers au moins un état stable qui n'est pas atteignable depuis les autres voies de différenciation. Dans l'hypothèse la plus forte, chaque voie de différenciation mène uniquement vers des états stables non atteignables par les autres voies de différenciation.

Définition 3.1.8 (réseau booléen compatible avec une bifurcation)**.** *On considère un point de bifurcation* p *ainsi que* o_1 *et* o_2 *deux observations appartenant à deux voies de différenciation différentes* v_1 *et* v_2 *issues de* p*. Étant donné une configuration* x *compatible avec* p *et deux configurations* d_1 *et* d_2 *respectivement compatibles avec* o_1 *et* o_2*, un réseau booléen* f *avec une sémantique* σ *est dit compatible avec la bifurcation* $\{p, o_1, o_2\}$ *associée aux voies* v_1 *et* v_2 *si et seulement si* $x \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} d_1$ *et* $x \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} d_2$ *et s'il existe des attracteurs* a_1 *et* a_2 *de* f *avec* σ *tels que* $d_1 \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} a_1$*,* $d_2 \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} a_2$*,* $d_1 \overset{f}{\underset{\sigma}{\nrightarrow}}{}^{*} a_2$ *et* $d_2 \overset{f}{\underset{\sigma}{\nrightarrow}}{}^{*} a_1$*.*

Un réseau booléen a une compatibilité *universelle* si l'ensemble des attracteurs atteignables depuis d_1 et l'ensemble des attracteurs atteignables depuis d_2 sont disjoints.

Définition 3.1.9 (réseau booléen compatible universellement avec une bifurcation)**.** *On considère un point de bifurcation* p *ainsi que* o_1 *et* o_2 *deux observations appartenant à deux voies de différenciation différentes* v_1 *et* v_2 *issues de* p*. Étant donné une configuration* x *compatible avec* p *et deux configurations* d_1 *et* d_2 *respectivement compatibles avec* o_1 *et* o_2*, un réseau booléen* f *avec une sémantique* σ *est dit compatible universellement avec la bifurcation* $\{p, o_1, o_2\}$ *associée aux voies* v_1 *et* v_2 *si et seulement si* $x \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} d_1$ *et* $x \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} d_2$*, et pour tout attracteur* a_1 *et* a_2 *de* f *avec la sémantique* σ*, si* $d_1 \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} a_1$ *et* $d_2 \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} a_2$ *alors* $d_1 \overset{f}{\underset{\sigma}{\nrightarrow}}{}^{*} a_2$ *et* $d_2 \overset{f}{\underset{\sigma}{\nrightarrow}}{}^{*} a_1$*.*

Exemple Le réseau booléen f avec la sémantique pleinement asynchrone dont le graphe de transitions est présenté en figure 3.6 est universellement compatible avec la bifurcation des observations B vers C et D présentées en figure 3.5. Cette compatibilité est visible sur le graphe de transition en figure 3.6 : depuis la configuration 1101 compatible avec B, sont atteignables les configurations 1000 et 0001 respectivement compatibles avec C et D et à partir desquelles les deux ensembles d'attracteurs atteignables, tous points fixes, sont bien disjoints : $\{\{1010\}\}$ depuis C et $\{\{0011\}, \{0101\}\}$ depuis D.

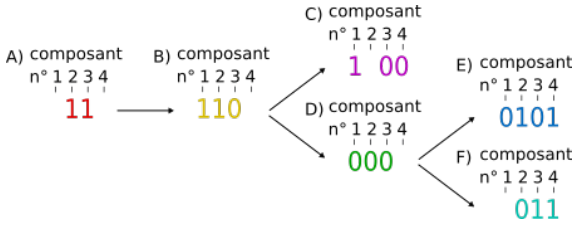


FIGURE 3.5 – Exemple d'observations organisées en un arbre de différenciation (déf. 3.1.21).

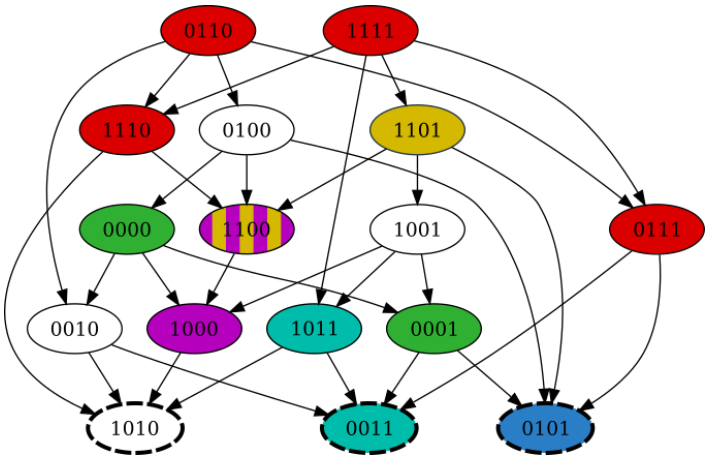


FIGURE 3.6 – Graphe de transitions du RB f défini en fig.2.1 en sémantique pleinement asynchrone, avec coloration des configurations compatibles avec les observations de même couleur présentées en fig.3.4. Les points fixes ont un contour en pointillés.

3.1.3 Stabilités cellulaires : marqueurs de stabilité partielle et totale

3.1.3.1 Information de stabilité associée à une observation

Le système qu'on observe peut avoir des composants qui n'évoluent pas, on parle alors de stabilité. Par exemple, dans le processus de différenciation des cellules du sang, on sait que le gène MS4A1 est constamment exprimé à partir du moment où la cellule est un lymphocytes B. La stabilité est ainsi une information qui provient des connaissances sur le contexte biologique et expérimental au moment de l'observation du système, par exemple parce que c'est une caractéristique connue sur le type cellulaire observé tel qu'un *marqueur* cellulaire ou parce qu'une perturbation expérimentale a été réalisée telle qu'une mutation.

Nous distinguons deux cas de figure d'informations de stabilités sur une observation. Suivant le contexte de l'observation, on peut souhaiter décrire spécifiquement quels sont les composants qui n'évolueront plus au sein du système, ou vouloir indiquer que c'est l'ensemble du système observé qui ne peut plus évoluer. Bien qu'intuitivement peu réaliste à l'échelle du système biologique, le second cas peut être considéré comme pertinent selon les données à disposition et l'ensemble de composants considéré pour décrire le système, par exemple lorsqu'on se focalise sur

les composants régulateurs du phénomène biologique observé.

3.1.3.2 Compatibilité avec un réseau booléen

Afin de faciliter la définition de compatibilité avec des informations de stabilité sur une observation, j'introduis un objet appelé *marqueur de stabilité* qui est l'association entre une observation et l'ensemble des composants du système qui sont stables.

Définition 3.1.10 (Marqueur de stabilité)**.** *On considère un système à n composants. Un marqueur de stabilité $s = (o, I)$ est un couple associant une observation $o \subseteq \{1, \ldots, n\} \times \mathbb{B}$ à un ensemble I de composants tel que $I \subseteq \{1, \ldots, n\}$. Un marqueur de stabilité est appelé* ***marqueur de stabilité partielle*** *lorsque $|I| < n$ et est appelé* ***marqueur de stabilité totale*** *lorsque $|I| = n$. Par souci de simplicité, un marqueur de stabilité totale peut être défini par l'observation seule puisque l'ensemble I est alors forcément égal à $\{1, \ldots, n\}$.*

J'ai mis en lien l'information de stabilité avec la propriété de confinement au sein d'un réseau booléen, confinement défini en 2.3.5. Lorsque cette information s'étend à l'ensemble des composants du système biologique observé, on est dans un cas particulier où on considère la propriété de point fixe au sein d'un réseau booléen, un point fixe correspondant à une configuration confinée sur l'ensemble des composants du système. À l'instar des listes d'observations et des bifurcations, il existe différentes interprétations possibles de la représentativité de l'information au regard du système observé. Suivant le contexte et les connaissances disponibles, on peut poser l'hypothèse faible que les états stables observés sont un aperçu des états stables possibles, ou bien supposer qu'ils correspondent à l'ensemble des états stables du système. J'ai donc décliné en deux définitions la compatibilité d'un réseau booléen avec des marqueurs de stabilité totale, la définition de base dite *existentielle* considérant l'hypothèse faible, la définition dite *universelle* considérant l'hypothèse forte.

- **Définitions existentielles :** Dans ce cadre, on considère l'existence du comportement voulu au sein de la dynamique du réseau booléen, sans contraindre l'ensemble de cette dynamique.

Définition 3.1.11 (réseau booléen compatible avec un marqueur de stabilité)**.** *Un réseau booléen f est dit compatible avec un marqueur de stabilité $s = (o, I)$ si et seulement s'il existe une configuration x compatible avec l'observation o qui est une configuration de f confinée sur I.*

Exemple Le réseau booléen f avec la sémantique pleinement asynchrone dont le graphe de transitions est présenté en figure 3.6 est compatible avec un marqueur de stabilité (partielle) $s = \{C, \{1, 4\}\}$ associant l'observation C de la figure 3.5 à l'ensemble de composants $I = \{1, 4\}$. En effet, premièrement 1100 est compatible avec C, et deuxièmement toute configuration atteignable depuis 1100, donc appartenant à $\rho_a^f(1100) = \{1000, 1010\}$, possède des valeurs identiques à 1100 pour les 1^{er} et 4^e composants (de valeurs respectivement 1 et 0).

Bien que la définition de compatibilité couvre les marqueurs de stabilité partielle et totale, nous privilégions la

définition suivante dans le cas particulier de la stabilité totale :

Définition 3.1.12 (réseau booléen compatible avec un marqueur de stabilité totale)**.** *Un réseau booléen f est dit compatible avec un marqueur de stabilité totale o si et seulement s'il existe une configuration x compatible avec l'observation o qui est un point fixe de f.*

En application, nous sommes amenés à considérer des informations de stabilité concernant un ensemble d'observations puisque différents états stables peuvent être observés. C'est donc la compatibilité avec un ensemble de marqueurs de stabilité qui sera considérée dans le cadre de la modélisation.

Définition 3.1.13 (réseau booléen compatible avec un ensemble de marqueurs de stabilité)**.** *Un réseau booléen f est dit compatible avec un ensemble S de marqueurs de stabilité si et seulement si, pour toute stabilité $s \in S$, f est compatible avec s.*

Exemple Le réseau booléen f avec la sémantique pleinement asynchrone dont le graphe de transitions est présenté en figure 3.6 est compatible avec l'ensemble de marqueurs de stabilité totale $S = \{E, F\}$, E et F correspondant aux observations de la figure 3.5. En effet, les configurations 0101 et 0011 sont respectivement compatibles avec E et F. f n'est pas compatible avec la stabilité étendue de $S' = \{C, E, F\}$ car il n'existe pas de point fixe compatible avec C.

- **Définitions universelles :** Pour étendre les applications possibles, il est intéressant de considérer également la possibilité d'imposer un comportement voulu dans tout ou partie de la dynamique du réseau booléen, en particulier dans une partie atteignable depuis une observation donnée. Dans le cas de la stabilité, le souhait est alors non seulement qu'il existe une compatibilité du réseau booléen avec l'ensemble des marqueurs de stabilité indiquées, mais également que tout attracteur du réseau booléen respecte au moins l'un des marqueurs de stabilité. Cette propriété permet d'imposer que tous les attracteurs du réseau booléen sont des états stables observés.

Définition 3.1.14 (réseau booléen compatible universellement avec un ensemble de marqueurs de stabilité)**.** *Un réseau booléen f est dit compatible universellement avec un ensemble S de marqueurs de stabilité si et seulement si, pour toute stabilité $s = (o, I) \in S$, f est compatible avec s et, pour tout attracteur a de f, il existe une configuration $x \in a$ telle que x est une configuration confinée de f sur I.*

Dans le cas particulier de la stabilité totale, nous privilégions la définition suivante :

Définition 3.1.15 (réseau booléen compatible universellement avec un ensemble de marqueurs de stabilité totale)**.** *Un réseau booléen f est dit compatible universellement avec un ensemble O de marqueurs de stabilité totale si et seulement si pour toute observation $o \in O$ il existe une configuration x compatible avec o qui est un point fixe et, pour tout x' point fixe de f, x' est compatible avec une observation $o' \in O$.*

Exemple Le réseau booléen f avec la sémantique pleinement asynchrone dont le graphe de transitions est présenté en figure 3.6 n'est pas universellement compatible avec l'ensemble de marqueurs de stabilité totale $O = \{E, F\}$

puisqu'il existe un point fixe, à savoir 1010 qui n'est compatible ni avec l'observation E, ni avec l'observation F. Ceci est vrai également en considérant l'ensemble $O' = \{C, E, F\}$: 1010 n'est pas non plus compatible avec C.

On peut souhaiter considérer l'universalité de la compatibilité uniquement sur une sous-partie de la dynamique d'un réseau booléen : la sous-partie atteignable depuis une observation donnée.

Définition 3.1.16 (réseau booléen compatible universellement avec un ensemble de marqueurs de stabilité atteignables depuis une observation donnée)**.** *On considère un réseau booléen f de dimension n avec une sémantique σ, une observation o et x une configuration compatible avec o, ainsi qu'un ensemble S de marqueurs de stabilité qui contient uniquement des observations atteignables depuis o. f est dit compatible universellement avec un ensemble S de marqueurs de stabilité atteignables depuis o si et seulement si, premièrement, pour tout marqueur de stabilité $s = (o', I) \in S$, il existe une configuration x' compatible avec o' telle que $x \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} x'$ et f est compatible avec s et, deuxièmement, pour tout attracteur a de f tel que $o \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} a$, il existe une configuration $x'' \in a$ et un marqueur de stabilité $s' = (o'', I') \in S$ tels que x'' est une configuration de f confinée sur I'.*

Dans le cas particulier de la stabilité totale, nous privilégions la définition suivante :

Définition 3.1.17 (réseau booléen compatible universellement avec un ensemble de marqueurs de stabilité totale atteignables depuis une observation donnée)**.** *On considère un réseau booléen f de dimension n avec une sémantique σ, une observation o et x une configuration compatible avec o, ainsi qu'un ensemble O de marqueurs de stabilité totale qui contient uniquement des observations atteignables depuis o. f est dit compatible universellement avec un ensemble O de marqueurs de stabilité totale atteignables depuis o si et seulement si, premièrement, pour toute observation $o' \in O$ il existe une configuration x' compatible avec o' telle que $x \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} x'$ et telle que x' est un point fixe, et deuxièmement, pour tout x'' point fixe de f tel que $x \overset{f}{\underset{\sigma}{\rightarrow}}{}^{*} x''$, x'' est compatible avec une observation $o'' \in O$.*

3.1.4 Différenciation cellulaire

Les observations d'une différenciation impliquent une combinaison des trois propriétés définies précédemment que sont les listes d'observations, les bifurcations et les stabilités. En effet, les données collectées sur un processus de différenciation sont par nature des observations ordonnées le long de l'évolution du processus. Afin de suivre un processus de différenciation, les observations ordonnées sont collectées sous la forme de plusieurs listes d'observations, le nombre de listes d'observations étant égal au nombre de voies différentes de différenciation possibles à partir du type cellulaire initialement observée. Les listes d'observations ont en commun des points de bifurcation à partir desquels naissent les différentes voies de différenciation, voies de différenciation qui peuvent être caractérisées par des informations sur la stabilité de tout ou partie des composants du système. La stabilité de l'ensemble des composants est usuellement utilisée pour décrire les états cellulaires stables en fin de différenciation.

Les données collectées sur un processus de différenciation peuvent être représentées sous la forme d'un *arbre de différenciation*. Afin de définir ce qu'est un arbre de différenciation, je rappelle des définitions de graphe et d'arbres en théorie des graphes.

Définition 3.1.18 (Graphe)**.** *Un graphe est un couple* (V, E) *tel que :*

- V *est un ensemble non vide quelconque dont les éléments sont appelés sommets ou nœuds.*
- $E \subset \{\{x, y\} \,|\, x, y \in V,\, x \neq y\}$*. Les éléments de* E *sont appelés les arêtes.*

Définition 3.1.19 (Arbre)**.** *Un arbre est un graphe* (V, E) *tel que :*

- *(Connexité)* $\forall x, y \in S$ *tels que* $x \neq y$, $\exists x_1, \ldots, x_n \in V$ *tels que* $\{x, x_1\}, \{x_1, x_2\}, \ldots, \{x_n, y\} \in E$*.*
- *(Acyclique)* $\forall x \in V$, $\nexists x_1, \ldots, x_n \in V$ *tels que* $\{x, x_1\}, \{x_1, x_2\} \ldots, \{x_n, x\} \in E$*.*

Vocabulaire associé :

- ***sommets adjacents*** *(deux sommets* x *et* y *sont adjacents s'ils sont reliés par une arête) :* $x, y \in V$ *tels que* $\{x, y\} \in E$*.*
- ***chemin*** *(liste de plusieurs sommets adjacents) :* $(v_1, \ldots, v_n)$ *tel que* $\forall i \in \{1, \ldots, n\}$*,* $v_i \in V$ *et tel que* $\forall i \in \{1, \ldots, n-1\}$*,* v_i *est adjacent à* v_{i+1}*.*
- ***distance entre deux sommets*** x ***et*** y *(taille du plus court chemin reliant* x *et* y*) : si* x *et* y *sont distincts alors la distance* $dist(x, y)$ *vaut* $\min\{n \geq 1 \,|\, \exists$ *chemin* $a_1, \ldots, a_n \in A$ *tel que* $x \in a_1$ *et* $y \in a_n\}$ *; sinon la distance vaut 0.*

L'arbre de différenciation que je souhaite définir est un type d'arbre particulier dit orienté et enraciné, également appelé *arborescence*. C'est un arbre dont toutes les arêtes sont dirigées vers l'extérieur à partir d'un sommet distingué appelé *racine*.

Définition 3.1.20 (Arbre enraciné orienté)**.** *Un arbre orienté enraciné* $\mathcal{A}$ *est un triplet* $(V, \hat{E}, r)$ *tel que :*

- $r \in V$ *et est appelé racine,*
- $\hat{E} \subset \{(x, y) \,|\, x, y \in V,\, x \neq y\}$*,*
- $\exists (x, y) \in \hat{E}$ *si et seulement si* $dist(r, x) < dist(r, y)$*,*
- *si on note* $E = \{\{x, y\} \,|\, (x, y) \in \hat{E}\}$ *alors* (V, E) *est un arbre.*

Remarque : pour tout sommet $v \in V$*, il y a exactement un seul chemin de* r *à* v*.*

Vocabulaire associé :

- ***sommets adjacents*** *(deux sommets* x *et* y *sont adjacents s'ils sont reliés par une arête dirigée de* x *vers* y*) :* $x, y \in V$ *tels que* $(x, y) \in E$*. On dit alors que* y *est* ***enfant*** *de* x *et que* x *est* ***parent*** *de* y*.*
- ***descendant d'un sommet*** *:* $desc(x) = y$ *si et seulement s'il existe un chemin de* x *vers* y*.*
- ***degré sortant d'un sommet*** *(nombre d'enfant du sommet) :* $d^+(x) = \#\{y \in V \,|\, (x, y) \in E\}$*.*
- ***feuille*** *(sommet dont le degré sortant vaut 0) :* $x \in V$ *tel que* $d^+(x) = 0$*.*

Définition 3.1.21 (Arbre de différenciation)**.** *Un arbre de différenciation $\mathcal{A} = (V, \hat{E}, r)$ est une arbre orienté enraciné tel que :*

- $\forall o \in V$*,* o *est une observation qui peut être associée à un marqueur de stabilité* (o, I) *;*
- *pour toute feuille* $z \in V$*, le chemin* $(r, \ldots, z)$ *est une liste d'observations dite incluse dans* $\mathcal{A}$ *;*
- $\forall p \in V$ *tel que* $d^+(p) > 1$*,* p *est un point de bifurcation inclus dans* $\mathcal{A}$ *avec, pour tout sommet* y *descendant de* p*,* y *préfixe d'une voie de différenciation issue de* p*. Étant donné* y_1 *et* y_2 *deux descendants de* p*,* (p, y_1, y_2) *est une bifurcation dite incluse dans* $\mathcal{A}$*.*

La figure 3.5 est un exemple d'arbre de différenciation.

Remarque : si on déclare une stabilité partielle avant une feuille, alors tout ce qui est après est inclus dans le trap space identifié (définition de trap space présentée en 2.2.5). La stabilité totale ne peut donc être associée qu'à des feuilles de l'arbre de différenciation.

L'arbre de différenciation est facilement généralisable en graphe orienté acyclique enraciné s'il y a besoin de davantage de flexibilité au sein des voies de différenciation pour décrire le comportement observé. L'information de bifurcation est alors associée à un nœud lorsque celui-ci a un degré sortant supérieur à 1 et possède des enfants n'ayant en commun aucun descendant. Ces enfants sont alors les préfixes des voies de différenciation issues de ce nœud.

Définition 3.1.22 (réseau booléen compatible avec un arbre de différenciation)**.** *Un réseau booléen f est dit compatible avec un arbre de différenciation $\mathcal{A} = (V, \hat{E}, r)$ si et seulement si :*

- *pour toute liste d'observations l incluse dans $\mathcal{A}$, f est compatible avec la liste d'observation l ;*
- *pour toute bifurcation b incluse dans $\mathcal{A}$, f est compatible avec la bifurcation b ;*
- *pour tout sommet o associé à un marqueur de stabilité $s = (o, I)$, f est compatible avec le marqueur de stabilité s.*

La compatibilité universelle avec l'arbre de différenciation peut être déclinée selon qu'on considère l'universalité sur les listes d'observations et/ou les bifurcations et/ou les marqueurs de stabilité. En présence de données de mutations, telles que pour la modélisation présentée dans [Chevalier et al., 2020], la nature des observations et de l'expérimentation rend pertinent une hypothèse forte sur les états stables observés qui sont atteignables depuis certaines observations. Une universalité peut de ce fait être ajoutée sur les marqueurs de stabilité totale et combinée aux informations d'ordre et de bifurcation pour décrire la différenciation cellulaire étudiée.

3.1.5 Comportements complexes de systèmes biologiques

Au-delà de la différenciation cellulaire formalisée ci-dessus, il est possible de décrire d'autres comportements complexes en généralisant la structure des données d'observations par un graphe orienté regroupant les informations

d'ordres, de stabilités et de bifurcations nécessaires à la description du comportement, informations déclinées selon les hypothèses faibles ou fortes. Au sein de ce graphe dont les nœuds sont des observations :

- l'information d'ordre est décrite par un arc dirigé entre deux nœuds,
- l'information de stabilité est décrite par son association avec un nœud,
- l'information de bifurcation est décrite par son association avec un nœud de degré sortant supérieur à 1 et ayant des enfants sans descendant commun. Ces enfants sont alors les préfixes des voies de différenciation issues de ce nœud.

Étant donné les nombreuses possibilités de combiner ces informations et les définitions de la compatibilité d'un réseau booléen avec chacune présentées en 3.1.1.2, 3.1.2.2 et 3.1.3.2, il est possible de considérer la compatibilité entre un réseau booléen et une grande diversité de comportements complexes afin d'aborder la modélisation de processus biologiques et expérimentaux variés.

3.2 Méthodes automatiques d'inférence de réseaux booléens

Il existe différentes méthodes pour inférer des réseaux booléens afin de modéliser des comportements biologiques. Ces méthodes diffèrent selon la stratégie mise en place pour aborder le problème d'inférence, mais elles diffèrent également par les données utilisées en entrée, les comportements qu'il est possible de modéliser et la façon dont ils sont considérés dans la dynamique (choix de sémantiques notamment), l'échelle des réseaux abordés et l'exploration de l'espace des solutions possibles. Je vais présenter un éventail des méthodes d'inférence, regroupées selon les comportements qu'elles peuvent modéliser.

Différents outils cherchent à modéliser les états stables du système observé, à partir d'un ensemble de mesures d'expression. Ainsi, les algorithmes génétiques proposés par CGA-BNI [Trinh and Kwon, 2021] et SgpNet [Gao et al., 2022], ou encore la récente méthode TaBooN [Aghamiri and Delaplace, 2021] couplant recherche tabou et résolution SAT, infèrent les réseaux booléens dont les points fixes respectent certains critères issus des données d'expression. CGA-BNI et SgpNet ne s'appuient pas sur un réseau de connaissances (*prior knowledge network* - PKN) pour construire les modèles : ce sont des outils qui infèrent le graphe d'interaction à partir des données.

Un ensemble d'autres outils abordent les trajectoires, en prenant en entrée des séries temporelles de mesures d'expression ou des données dites de *perturbation* recueillies sous différentes conditions. Parmi ceux-ci se trouvent notamment CellNOptR [Terfve et al., 2012] qui, en sémantique synchrone, via la programmation linéaire en nombres entiers, considère les points fixes et définit une contrainte d'atteignabilité dans le cas des données de perturbation. BONITA [Palli et al., 2019] étend cette modélisation à la sémantique asynchrone. L'inférence proposée par ASKEed [Vaginay et al., 2021] (programmation par ensemble-réponse - ASP), modélise en asynchrone les séries temporelles sous la forme de trajectoires au sein desquelles la succession de mesures constitue une succession de transitions

(atteignabilité en un seul pas de dynamique). Caspo-ts [Ostrowski et al., 2016a], également en asynchrone, relâche la contrainte d'observer chaque transition et considère directement la propriété d'atteignabilité entre les configurations observées. Enfin, contrairement aux méthodes citées jusqu'ici dans ce paragraphe qui considèrent toutes un PKN en entrée, il existe des méthodes qui infèrent le graphe d'interaction directement depuis les données, comme notamment la méthode GAPORE [Liu et al., 2021] (algorithme génétique).

Un outil se distingue afin d'aborder des comportements plus complexes : BRE :IN [Goldfeder and Kugler, 2019]. La richesse des comportements pouvant être considérés n'est égalée par aucune autre méthode étant donné que cet outil peut vérifier les propriétés exprimées en logique temporelle, tant en sémantique synchrone qu'asynchrone. Cependant, la vérification de ces traces en logique temporelle est très coûteuse, ce qui empêche le passage à l'échelle de réseaux d'interactions biologiques de plusieurs dizaines de composants.

Plus généralement, le passage à l'échelle est difficile pour l'ensemble des méthodes citées jusqu'ici. Un autre défi important est l'exploration de l'espace des modèles possibles. En effet, la grande majorité des méthodes se basent sur des heuristiques améliorant itérativement un modèle initial dont les règles logiques sont prédéfinies. Le modèle initial considère habituellement comme point de départ l'inhibiteur comme dominant, c'est-à-dire que même si l'ensemble des activateurs d'un composant sont actifs, l'activation d'un seul inhibiteur suffit à inactiver le composant. Avec BoNesis, nous proposons une approche logique de l'inférence, exprimée comme un problème de satisfiabilité, qui permet d'accéder à l'ensemble complet des modèles possibles tout en passant à l'échelle de réseaux de plusieurs milliers de nœuds (je présente au chapitre 5 une application avec un graphe d'interactions initial de plus de 1000 composants). Les contraintes, écrites en ASP, permettent de modéliser des comportements biologiques complexes tels que la différenciation cellulaire. En effet, BoNesis prend en compte la compatibilité d'un réseau booléen avec des listes d'observations mais également avec des bifurcations entre ces listes d'observations. De plus, il considère plusieurs propriétés de stabilité afin de représenter les états stables via la stabilité de l'ensemble des composants du système, tel que considéré par les méthodes existantes, ou de seulement une partie des composants, afin de décrire la stabilité de marqueurs biologiques par exemple. De ce fait, BoNesis peut autant aborder des mesures d'expression en séries temporelles et à l'état stable que des données de perturbations. Pour finir, notre méthode calcule la dynamique selon la sémantique *Most Permissive* afin de ne pas manquer de trajectoire pouvant faire sens biologiquement, tout particulièrement des trajectoires résultant de seuils d'activation différents entre cibles d'un même composant et qui ne peuvent être reproduites en asynchrone.

Résumé du chapitre 3

Je propose un cadre formel sur les propriétés des données collectées pour étudier des comportements biologiques tels que la différenciation cellulaire. J'ai distingué trois propriétés majeures qui peuvent être combinées pour décrire des comportements biologiques complexes, et j'ai décomposé la compatibilité d'un réseau booléen selon deux interprétations possibles de chacune de ces propriétés. En effet, on peut souhaiter un modèle qui reproduise le comportement observé en excluant, ou non, d'autres comportements possibles. Ainsi, soit il est suffisant d'assurer que le comportement observé est inclus dans la dynamique du modèle, interprétation de la compatibilité que je nomme *existentielle*, soit il faut garantir une *universalité* du comportement au sein de la dynamique, compatibilité universelle que je définis selon les propriétés abordées étant donné les besoins des modélisateurs.

En premier lieu, je rappelle que l'information recueillie à partir d'observations du système biologique fournit une information binarisée indiquant pour un ensemble de composants s'ils sont présents ou absents, exprimés ou inhibés, actifs ou inactifs. De ce fait, j'ai formalisé une observation comme étant un ensemble au sein duquel sont associés des composants avec une valeur booléenne. L'information d'ordre entre les observations est importante puisque cet ordonnancement permet de décrire l'évolution d'un état cellulaire et donc des trajectoires. Je représente cette information par une liste d'observations et je formalise la compatibilité d'un réseau booléen avec une liste d'observations, sous une interprétation existentielle et universelle.

Je formalise également l'existence d'évolutions différentes à partir d'un même état initial en définissant une bifurcation associée à des listes d'observations. La bifurcation implique un point de bifurcation ainsi que deux voies de différenciation qui ne peuvent se rejoindre. Je définis la compatibilité existentielle et universelle d'un réseau booléen avec une telle bifurcation. La définition universelle répond tout particulièrement au besoin de modéliser l'observation exhaustive de phénotypes atteints lorsque des cellules sont soumises à différentes conditions expérimentales, spécifique de l'étude de l'impact de différentes mutations par exemple.

Enfin, pour aborder la possibilité d'atteindre des états cellulaires stables, j'ai distingué deux cas de figure d'informations de stabilité sur une observation. Dans un cas, on souhaite modéliser la stabilité connue de quelques composants, typiquement les gènes utilisés comme marqueurs de type cellulaire dont l'expression est caractéristique une fois la cellule différenciée. Dans l'autre cas, c'est l'ensemble du système de régulation qui est considéré comme stable, modélisé par la stabilité de l'ensemble des composants constituant le modèle. Pour cela, je représente la stabilité d'une observation via un *marqueur de stabilité* qui peut être partielle ou totale, que j'associe respectivement aux propriétés de confinement ou de point fixe au sein d'un réseau booléen compatible.

En réponse à ces besoins de la modélisation en biologie, des méthodes d'inférence automatique de réseaux booléens ont été développées. Je réalise un état de l'art de ces approches automatiques afin de montrer les propriétés dynamiques qu'elles abordent au regard des besoins en biologie. Je souligne, au regard de ces méthodes, les apports de celle que je développe dans la thèse. Il est important de noter que l'inférence automatique ne fait sens qu'avec une implication manuelle experte afin de spécifier les observations et interpréter le comportement observé.

Chapitre 4

Encodage en programmation par ensemble-réponse

Sommaire

Une méthode d'inférence de réseaux booléens à partir de propriétés positives d'atteignabilité, basée sur la programmation par ensemble-réponse (*Answer-Set Programming* (ASP)) et sur un filtrage des solutions par model-checking avec la sémantique asynchrone, avait été mise au point dans le cadre d'une collaboration entre Loïc Paulevé, Anne Siegel (IRISA, Rennes), Carito Guziolowsky (LS2N, Nantes) et Max Ostrowski (Univ. Potsdam, Allemagne) [Ostrowski et al., 2016b]. Sur la base de ce travail, j'ai implémenté la vérification de propriétés dynamiques supplémentaires, avec la sémantique *Most Permissive* (MP), afin de rendre possible la modélisation de comportements biologiques complexes tels que la différenciation cellulaire.

J'ai pour cela créé des contraintes d'existence de points fixes et d'irréversibilité d'une bifurcation afin de reproduire les propriétés observées d'une population cellulaire en différenciation, c'est-à-dire évoluant vers des destins mutuellement exclusifs. Ces contraintes, ainsi que la présentation de la stratégie d'utilisation d'ASP pour parvenir à l'inférence de modèles de réseaux de régulation biologique, ont été publiées dans [Chevalier et al., 2019], contribution que j'ai présentée à la conférence ICTAI 2019. J'ai également abordé un nouveau type de propriétés appelées *universelles*, afin de ne pas seulement assurer qu'un comportement est dans la dynamique du système, mais aussi garantir que c'est le seul comportement possible. Pour illustrer ce point, comparons la signification des propriétés existentielles et universelles à partir d'une liste de destins cellulaires observés expérimentalement : une contrainte existentielle garantit qu'au moins un attracteur de la dynamique du modèle correspond à chaque destin cellulaire, tandis qu'une contrainte universelle garantit que chaque attracteur du modèle correspond à au moins un destin cellulaire. J'ai ainsi étendu la contrainte de point fixe afin de l'aborder de façon universelle, soit à l'échelle de l'ensemble de la dynamique du réseau booléen, soit au sein d'une sous-partie de cette dynamique limitée par une propriété d'atteignabilité. Cette contrainte a été publiée dans [Chevalier et al., 2020] au sein d'un travail que j'ai présenté à la conférence CMSB 2020.

Dans ce chapitre, je détaille tout d'abord la stratégie utilisée pour aborder le problème d'inférence de modèles. Je poursuis en présentant les bases de la programmation par ensemble-réponse utilisée pour l'encodage de l'inférence de modèles. La dernière partie du chapitre est dédiée à cet encodage avec, en premier lieu, l'implémentation des fonctions booléennes et de leur évaluation. Je présente ensuite la façon dont j'ai implémenté les contraintes existentielles d'atteignabilité positive et d'atteignabilité négative, qui garantissent respectivement l'existence et l'absence de trajectoire d'une configuration à une autre, ainsi que les contraintes existentielles de point fixe et de confinement, afin de garantir respectivement la stabilité de tout ou partie des composants d'une configuration. Le chapitre se termine sur la présentation de la contrainte universelle sur les points fixes, contrainte limitée ou non par une propriété d'atteignabilité.

4.1 Principe de la méthode

Nous formulons le problème de l'inférence de réseaux booléens comme un problème de satisfiabilité booléenne codé en Answer-Set Programming. Le problème d'inférence est décrit sous forme d'un programme logique en Answer-Set Programming qui contient à la fois la description du formalisme du modèle et les données du processus biologique à modéliser. Avec cette approche, les connaissances et les données expérimentales sont exploitées comme contraintes sur la topologie du graphe d'interactions et sur les propriétés dynamiques des réseaux booléens (sous la sémantique MP). J'ai développé des contraintes afin de faire correspondre les propriétés dynamiques des réseaux booléens avec les comportements biologiques observés sur des cellules évoluant vers des phénotypes mutuellement exclusifs.

Le programme logique qui est créé pour inférer les modèles intègre :

- la définition de réseau booléen localement monotone,
- le calcul de sa dynamique selon la Most Permissive Semantics (étant donné les garanties apportées et sa faible complexité),
- le PKN, qui définit le domaine des fonctions booléennes possibles,
- les données biologiques associées à des contraintes décrivant leurs propriétés dynamiques.

Une solution du programme sera donc un réseau booléen qui (i) appartient au domaine défini par le PKN et (ii) dont la dynamique MP respecte les propriétés associées aux données.

Les propriétés dynamiques à respecter, qui diffèrent selon la nature des données biologiques et le processus étudié, sont implémentées sous la forme de contraintes que la dynamique du réseau booléen doit respecter. Plus l'éventail de propriétés prises en compte par la méthode d'inférence est large, plus il est possible de décrire finement la dynamique des données et donc de modéliser des phénomènes complexes.

4.1.1 Formalisation du problème d'inférence de modèle

Une observation (partielle) o d'une configuration de dimension n est spécifiée par un ensemble de couples associant un composant à une valeur booléenne : $o \subseteq \{1, \ldots, n\} \times \mathbb{B}$, supposant qu'il n'y a aucun $i \in \{1, \ldots, n\}$ tel que $\{(i, 0), (i, 1)\} \subseteq o$.

Formellement, le problème d'inférence que nous abordons est le suivant. Étant donné :

- un graphe d'interactions $\mathcal{G} = (\{1, \ldots, n\}, E_+, E_-)$;
- p observations partielles $o^1, \ldots, o^p$;
- des ensembles PR, URFP, NR de couples d'indices d'observations :
  $\mathsf{PR}, \mathsf{URFP}, \mathsf{NR} \subseteq \{1, \ldots, p\}^2$;

- des ensembles FP et UFP d'indices d'observations :

  $\mathsf{FP}, \mathsf{UFP} \subseteq \{1, \ldots, p\}$;
- un ensemble TP associant des indices d'observations avec des composants :

  $\mathsf{TP} \subseteq \{1, \ldots, p\} \times \{1, \ldots, n\}$.

Le problème consiste à trouver un réseau booléen f de dimension n tel que :

- $G(f) \subseteq \mathcal{G}$,
- il existe p configurations $x^1, \ldots, x^p$ telles que :
  - **observations :** $\forall m \in \{1, \ldots, p\}, \forall (i, v) \in o^m, x_i^m = v$ (pour toute observation d'indice m, chaque composant i observé dans l'observation o^m a la même valeur v dans la configuration compatible x^m) ; afin de donner l'ensemble des valeurs des composants observés, au sein d'observations du système telles que définies en 3.1.1.
  - **atteignabilités positives, *Positive Reachability (PR)* :** $\forall (m, m') \in \mathsf{PR}, x^m \rightarrow^* x^{m'}$ (pour tout couple d'observations d'indices m et m' présent dans l'ensemble PR, il existe un chemin d'une configuration x^m compatible avec o^m vers une configuration $x^{m'}$ compatible avec $o^{m'}$) ; afin de décrire l'évolution du système observé via une information d'ordre entre observations. On assure un réseau booléen compatible avec les listes d'observations contenues dans PR.
  - **atteignabilités négatives, *Negative Reachability (NR)* :** $\forall (m, m') \in \mathsf{NR}, x^m \not\rightarrow^* x^{m'}$ (pour tout couple d'observations d'indices m et m' présent dans l'ensemble NR, il n'existe pas de chemin d'une configuration compatible x^m vers une configuration compatible $x^{m'}$) ; afin de décrire qu'à partir d'une configuration compatible avec une observation o^m, une configuration compatible avec une observation $o^{m'}$ ne peut pas être atteinte. Avec les atteignabilités négatives contenues dans NR, on assure en partie la compatibilité d'un réseau booléen avec une bifurcation (déf. 3.1.8) en décrivant l'impossibilité, à partir d'une voie de différenciation issue d'un point de bifurcation p, d'atteindre une autre voie issue de p.
  - **points fixes, *Fixed Point (FP)* :** $\forall m \in \mathsf{FP}, f(x^m) = x^m$ (pour toute observation d'indice m présente dans l'ensemble FP, il existe une configuration compatible x^m qui est un point fixe) ; afin de décrire la stabilité totale de certaines observations (déf. 3.1.10). On assure un réseau booléen compatible avec l'ensemble des marqueurs de stabilité totale contenus dans FP, tel que défini en 3.1.13.
  - **confinements, en lien avec la notion de *Trap Space (TP)* :** $\forall (m, i) \in \mathsf{TP}, \exists t \in (\mathbb{B} \cup \{*\})^n$ tel que t est le plus petit *trap space* contenant x^m et $t_i = x_i^m$ (pour toute observation d'indice m associée à un composant d'indice i dans l'ensemble TP, il existe un hypercube t tel que t est le plus petit trapspace contenant une configuration compatible x^m et tel que le composant i de t a la même valeur que dans l'observation o^m) ; afin de décrire la stabilité de certaines observations sur un ensemble de composants, c'est-à-dire l'existence d'attracteurs, potentiellement cycliques, où ces composants ont leur valeur fixe (marqueur de stabilité partielle défini en 3.1.10). On assure un réseau booléen compatible avec l'ensemble des

marqueurs de stabilité contenus dans TP.

– **points fixes universels, *Universal Fixed Point (UFP)* :** $\forall z \in \mathbb{B}^n, f(z) = z \Rightarrow \exists m \in \text{UFP} : \forall (i,v) \in o^m, z_i = v$ (pour toute configuration z, si z est un point fixe alors il existe un indice d'observation m dans l'ensemble UFP tel que tout composant d'indice i observé dans l'observation o^m a la même valeur dans z) ; afin de décrire que tous les états stables du système correspondent aux stabilités totales observées. On assure un réseau booléen compatible universellement avec l'ensemble des marqueurs de stabilité totale contenus dans UFP.

– **points fixes universels atteignables, *Universal Reachable Fixed Point (URFP)* :** $\forall (q, m') \in \text{URFP}, \forall z \in \mathbb{B}^n, (f(z) = z \wedge x^q \rightarrow^* z) \Rightarrow \exists (q,m) \in \text{URFP} : \forall (i,v) \in o^m, z_i = v$ (pour tout couple d'indices d'observation q et m' dans l'ensemble URFP et pour toute configuration z, si z est atteignable depuis une configuration compatible x^q et que z est un point fixe alors il existe un couple d'indices d'observation q et m dans URFP tel que tout composant d'indice i observé dans l'observation o^m a la même valeur dans z) ; afin de spécifier l'ensemble des stabilités totales auxquelles doivent correspondre les états stables atteints à partir d'une observation donnée. On assure un réseau booléen compatible universellement avec l'ensemble des marqueurs de stabilité totale atteignables contenus dans URFP.

On peut éventuellement imposer également que le graphe d'interactions de f soit égal à l'entrée $\mathcal{G}$.

Notons qu'un tel problème peut être non satisfiable selon le graphe d'interactions et les propriétés dynamiques données.

Exemple Nous pouvons aborder le problème d'inférence de réseaux booléens compatible avec le graphe d'interactions présenté en figure 2.2 et avec le comportement en figure 3.5. La figure n'associant aucun marqueur de stabilité aux observations, je suppose que chaque feuille est associée à un marqueur de stabilité totale (déf. 3.1.10) considéré selon la définition existentielle. Le problème consiste alors à trouver un réseau booléen f de dimension 4 tel que :

- le graphe d'interactions est inclus dans $\mathcal{G} = (\{1,2,3,4\}, E_+ = \{(4,2)\}, E_- = \{(1,2),(2,3),(3,2),(1,4),(4,1)\})$;
- il existe l'ensemble d'observations $p = \{A, B, C, D, E, F\}$ tels que $A = \{(2,1),(3,1)\}$, $B = \{(1,1),(2,1),(3,0)\}$, $C = \{(1,1),(3,0),(4,0)\}$, $D = \{(1,0),(2,0),(3,0)\}$, $E = \{(1,0),(2,1),(3,0),(4,1)\}$ et $F = \{(2,0),(3,1),(4,1)\}$;
- il existe l'ensemble d'atteignabilités positives $\text{PR} = \{(A,B),(B,C),(B,D),(D,E),(D,F)\}$ (flèches de la fig. 3.5) ;
- il existe l'atteignabilité négative $\text{NR} = \{(D,C)\}$ décrivant l'impossibilité d'atteindre C depuis D ;
- il existe l'ensemble de points fixes $\text{FP} = \{C, E, F\}$ décrivant la stabilité totale des observations finales.

Nous pouvons également souhaiter décrire la stabilité selon la définition universelle en précisant que, d'une part, les points fixes doivent être compatibles avec C, E ou F et que, d'autre part, tout point fixe atteignable depuis D est compatible soit avec E soit avec F. Dans ce cas nous ajoutons :

- la contrainte de points fixes universels : $\text{UFP} = \{C, E, F\}$,

- la contrainte de points fixes universels atteignables depuis D : URFP $= \{(D, E), (D, F)\}$.

On remarque que l'ensemble NR de contraintes d'atteignabilités négatives devient alors inutile puisque URFP implique qu'aucun point fixe atteignable depuis D ne peut être compatible avec C.

Énumération exhaustive des solutions Notre implémentation évite la redondance entre modèles en n'énumérant que les réseaux booléens non équivalents, c'est-à-dire dont les valeurs diffèrent pour au moins une configuration. C'est réalisé grâce à une représentation canonique des fonctions booléennes sous forme normale disjonctive avec un ordre total entre les clauses.

4.2 Answer-Set Programming

La programmation par ensemble-réponse (ASP) est une approche logique déclarative pour résoudre des problèmes combinatoires de satisfiabilité. Elle est proche de SAT (satisfiabilité propositionnelle) et ses solveurs sont connus pour être efficaces pour l'énumération de solutions de problèmes NP ayant jusqu'à des dizaines de millions de variables. Dans cette section, j'introduis les bases du langage ASP afin d'expliquer la syntaxe que nous utilisons dans les prochaines sections, et je présente la sémantique des *modèles stables* sur laquelle repose ASP.

4.2.1 Atomes

Les programmes logiques sont construits à partir d'atomes. Un atome est constitué par un prédicat dont les arguments sont des termes.

Les types de termes les plus simples sont les entiers, les constantes et les variables. Constantes et variables se distinguent par leur première lettre, respectivement une minuscule et une majuscule. Un terme plus complexe est formé d'une fonction, désignée par un nom de fonction commençant par une minuscule, avec un ou plusieurs termes passés en arguments ; le nombre d'arguments est l'arité de la fonction. Par exemple, `X` et `anna` sont des termes, ainsi que `pere(anna)` et `pere(mere(anna))`. Un terme qui ne contient aucune variable est appelé un terme de base (*ground term*).

Un atome est une construction élémentaire pour représenter la connaissance, elle renvoie à une affirmation verbale élémentaire. Par exemple, `rencontre(anna,mere(anna),X)` est un atome constitué du symbole de prédicat `rencontre` d'arité 3 avec pour arguments : la constante `anna`, le terme `mere(anna)` construit avec le symbole de fonction unaire `mere`, et la variable `X` désignant un lieu quelconque. Un atome qui ne contient aucune variable est appelé un *ground atom*. `rencontre(anna,mere(anna),X)` n'est pas un ground atom, tandis que `rencontre(anna,mere(anna),Perpignan)` en est un.

4.2.2 Règles et dérivation

Les règles d'un programme en ASP suivent la forme suivante :

Fait : H.

Règle : $H \leftarrow L_1, \ldots, L_n$.

Contrainte d'intégrité : $\leftarrow L_1, \ldots, L_n$.

où $H, L_1, \ldots, L_n$ sont des littéraux, constituant la *tête* et le *corps* de la règle, avec la particularité que le fait est une règle constituée uniquement d'une tête, et qu'une contrainte d'intégrité est une règle ayant uniquement un corps. Dans le corps d'une règle ou d'une contrainte d'intégrité, chaque L_j pour $1 \leq j \leq n$ est un littéral de la forme `A` ou not `A`, où `A` est un *atome* et le connecteur logique not désigne la négation par défaut. On dit qu'un littéral L est positif si c'est un atome (`A`), et négatif sinon (not `A`). Une solution du programme logique, aussi appelée un *modèle*, est un ensemble d'atomes. L'ASP étant basé sur la sémantique des modèles stables présentée en sous-section 4.2.4, par solution on désigne un modèle particulier appelé *modèle stable* ou *ensemble-réponse*.

Un fait, qui est une règle sans corps, implique que l'atome de tête H doit obligatoirement être vrai. Une règle, quant à elle, correspond à une implication : intuitivement, la tête d'une règle est vraie si tous les littéraux de son corps sont vrais. Ainsi, si tous les littéraux positifs du corps de la règle sont vrais et que tous les littéraux négatifs sont satisfaits (c'est-à-dire qu'aucun d'eux ne peut être dérivé), alors H doit être vrai. En ASP, pour être vrai, un atome doit pouvoir être dérivé, c'est-à-dire qu'il ne pourra jamais être vrai s'il ne figure dans la tête d'aucune règle. À noter, la dérivation doit être acyclique. Pour illustrer cela, considérons le programme logique suivant :

```
1 a ← b.
2 b ← a.
```

L'ensemble vide est l'unique solution de ce programme. En effet, l'ajout de `a` ou de `b` engendre forcément l'ensemble $\{a, b\}$, puisque `a` et `b` ne peuvent pas être dérivés de façon acyclique (`a` dépend de `b`, mais `b` dépend de `a`). Enfin, une contrainte d'intégrité est une règle qui élimine les ensemble-réponses candidats si l'ensemble des littéraux de son corps sont satisfaits. Elle constitue ainsi un "test" écartant les ensemble-réponses non souhaités. Par exemple, considérons la contrainte d'intégrité suivante :

```
3 ← a, not b.
```

Pour qu'un ensemble-réponse soit valide, cette règle implique que l'atome `a` en soit absent (ce qui implique que `a` n'a pu être dérivé par aucune autre règle du programme logique), ou que l'atome `b` y soit présent (`b` ayant alors été dérivé par une autre règle).

Pour qu'un ensemble d'atomes constitue l'ensemble-réponse d'un programme logique ASP, il doit satisfaire toutes les règles, faits et contraintes d'intégrité de ce programme.

4.2.3 Notations

Je présente ci-dessous quelques notations permises par le langage qui ont été utilisées pour l'encodage des contraintes présentées à la section 4.3.

- `a((x;y))` qui est développé en `a(x), a(y).`
- `#count {X: a(X)}` qui est le nombre de `X` distincts pour lesquels `a(X)` est vrai.
- n `{a(X): b(X)}` m qui est vrai si au moins n et au plus m atomes `a(X)` sont vrais, avec `X` appartenant aux `b(X)` vrais.
- `a(X): b(X)` qui est vrai si pour chaque `b(X)` vrai, `a(X)` est vrai.

Si dans le corps de la règle une telle condition est suivie d'un terme, alors condition et terme sont séparés par `;`.

Il est également possible d'introduire un choix dans la tête de la règle. Par exemple :

- `{a}` $\leftarrow$ *body*. laisse au solveur la possibilité de mettre `a` à vrai si le corps de la règle est satisfait ;
- `{a,b,c}` $\leftarrow$ *body*. laisse au solveur la possibilité de mettre soit `a`, soit `b`, soit `c` à vrai si le corps de la règle est satisfait.

Le langage ASP permet également d'utiliser des variables (nommées en commençant par une majuscule) qui vont être remplacées par des termes lors de l'étape de *grounding* afin d'obtenir des atomes sans variables.

Par exemple, le programme ASP suivant contenant la variable `X` :

```
4 a(X) ← b(X).
5 b(1).
6 b(2).
```

engendre l'instanciation de `X` par `1` et `2` ce qui aboutit, après *grounding*, au programme suivant :

```
7 a(1) ← b(1).
8 a(2) ← b(2).
9 b(1).
10 b(2).
```

4.2.4 Modèle stable

Une solution d'un programme logique en ASP est appelée un *ensemble-réponse* ou un *modèle stable* [Gelfond and Lifschitz, 1988]. En effet, la sémantique des *modèles stables* est à la base du langage ASP. Elle utilise la *négation par défaut* qui considère une représentation complète de l'état des connaissances, c'est-à-dire que les atomes qui appartiennent à l'ensemble-réponse sont connus pour être vrais, et les atomes qui n'appartiennent pas à l'ensemble-réponse sont considérés comme faux. Ainsi, la négation par défaut implique que s'il y a échec de la dérivation de `p`, alors not `p` est dérivé.

Afin de décrire ce qu'est un modèle stable, considérons P un ensemble de règles de la forme :

11 $H \leftarrow B_1, \ldots, B_m, \text{ not } C_1, \ldots, \text{ not } C_n$

avec $H, B_1, \ldots, B_m, C_1, \ldots, C_n$ des *ground atoms*.

Si P ne contient pas de négation Un modèle stable est construit comme suit. Tout d'abord, l'ensemble des atomes qui sont des faits y sont intégrés. Ensuite, de façon itérative jusqu'à ce que l'ensemble cesse de croître, s'y ajoutent les atomes présents dans la tête de règles dont le corps est satisfait par les atomes déjà présents dans le modèle. Si cet ensemble d'atomes n'est pas éliminé par une contrainte d'intégrité, alors cet ensemble constitue un modèle stable de P.

S'il n'y a de choix dans aucune tête de règles dans P, il existe alors un unique modèle stable. Alors que dans le cas où la tête d'une règle permet un choix, il peut exister autant de modèles stables que de choix possibles.

Si P est un programme avec négation On a besoin d'appliquer le concept de *réduction* qui se définit comme suit. Pour tout ensemble I d'atomes de base, la réduction de P par rapport à I est l'ensemble des règles sans négation obtenu à partir de P en supprimant :

1. chaque règle ayant au moins un des littéraux négatifs de son corps (atomes C_i) qui appartient à I,
2. les littéraux négatifs du corps de toutes les règles restantes.

On dit alors que I est un modèle stable de P si I est le modèle stable du programme P_I obtenu en réduisant P par rapport à I. Par construction, P_I ne contient pas de négation et son modèle stable est donc défini comme décrit au paragraphe précédent. Un programme P avec négation peut avoir plusieurs modèles stables ou bien aucun ; il se peut en effet que plusieurs ensembles I différents soient solutions du programme réduit P_I, ou bien aucun.

Afin d'illustrer la réduction, considérons deux ensemble-réponses candidats pour le programme logique suivant :

12 `a.`

13 `c` $\leftarrow$ `a, b.`

14 `d` $\leftarrow$ `a,` not `b.`

- Pour calculer si $I = \{a, d\}$ est un ensemble-réponse, la réduction du programme étant donné I donne :

15 `a.`

16 `c` $\leftarrow$ `a, b.`

17 `d` $\leftarrow$ `a.`

En effet, dans un premier temps, aucune règle n'a été entièrement supprimée puisqu'aucune ne contient, dans son corps, un littéral négatif dont l'atome est a ou d. Dans un second temps, le littéral négatif de la $3^{\text{è}}$ règle a été supprimée. Ce programme réduit a pour modèle stable $\{a, d\}$, cet ensemble est donc bien un modèle stable du programme originel avec négation.

- Pour calculer si $I = \{a, b, c\}$ est un ensemble-réponse, la réduction du programme étant donné I donne :

```
18 a.
19 c ← a, b.
```

En effet, la $3^{\text{è}}$ règle a été supprimée puisqu'elle contenait le littéral négatif not `b` alors que l'atome `b` appartient à I. Aucun littéral n'a ensuite été supprimé puisque les règles restantes ne contiennent aucun littéral négatif. Ce programme réduit a pour modèle stable $\{a\}$. Puisqu'il diffère de I, I n'est pas un modèle stable du programme originel avec négation.

4.2.4.1 Propriétés de la sémantique des modèles stables

En se basant sur la sémantique des modèles stables, ASP a les propriétés suivantes :

Minimalité Un modèle stable d'un programme logique P est un modèle minimal parmi les modèles de P, au sens de l'inclusion des ensembles. Ainsi, si un programme a plusieurs modèles stables, aucun d'eux n'est inclus dans un autre puisque chacun est minimal.

NP-complétude Déterminer si un programme logique a un modèle stable est NP-complet.

4.2.5 Règles disjonctives

ASP peut exprimer des programmes logiques dits *disjunctifs* [Lobo et al., 1992] au moyen de disjonctions dans la tête de la règle (atomes séparés par ";"). Une règle disjonctive est de la forme :

```
20 a; b ← body.
```

La règle suivante est un exemple de règle disjonctive :

```
21 innocent(X); guilty(X) ← suspect(X).
```

Une telle règle dit qu'au moins un des éléments de la tête de la règle doit être vrai. Or la sémantique ASP implique le critère de minimalité qui ici implique qu'un ensemble réponse est une solution seulement si aucun de ses sous-ensembles est lui-même une solution [Eiter et al., 2009]. Par exemple, considérons la disjonction suivante :

```
22 a; b; c.
```

L'interprétation $I = \{a, b\}$ est un modèle mais elle n'est pas minimale. En effet, les interprétations $\{a\}$ et $\{b\}$ sont strictement incluses dans I et vérifient également la règle. Par conséquent, I n'est pas une solution.

En utilisant des règles disjonctives, il a été montré dans [Eiter and Gottlob, 1995] que la complexité des problèmes traités avec ASP peut être étendue aux formules booléennes quantifiées d'ordre 2 (*2QBF*, c'est-à-dire de la forme $\forall x \exists y.\phi$ ou $\exists y \forall x.\phi$, avec ϕ une formule sans quantificateur). La satisfiabilité d'une formule 2QBF peut être réduite à la vérification de l'existence d'un ensemble-réponse pour un programme logique disjonctif en ASP.

4.2.6 Résolution

L'Answer-Set Programming permet de décrire de façon simple mais puissante un problème combinatoire sous la forme d'un programme logique. Parmi les outils mis à disposition autour d'ASP pour exploiter des programmes logiques, nous avons choisi l'outil *clingo* [Gebser et al., 2014]. Cet outil combine en fait les fonctionnalités de *gringo* et *clasp*, deux outils indispensables et complémentaires pour la résolution. Le premier, gringo, est un *grounder*, c'est-à-dire un logiciel qui "traduit" le programme logique fourni par l'utilisateur, qui contient des variables, en un programme logique équivalent sans variable (voir l'exemple associé au *grounding* en section 4.2.3). Le deuxième, clasp, est un *solveur* qui calcule les ensemble-réponses du programme produit par gringo. Ainsi, à partir d'un programme logique fourni par l'utilisateur en entrée, clingo énumère l'ensemble des solutions. Il offre différentes fonctionnalités dont les principales sont la *projection*, l'*optimisation*, et la possibilité d'intégrer des heuristiques dans la résolution ASP, que nous exploitons pour la méthode d'inférence de modèles de réseaux biologiques :

Projection Il est habituel que la solution soit caractérisée par seulement un sous-ensemble des atomes appartenant à l'ensemble-réponse. C'est pourquoi il est possible de préciser une projection dans le programme logique par l'instruction `#show`, afin de cacher l'ensemble des atomes non pertinents. Pour illustrer, reprenons l'exemple en 4.2.3 en y ajoutant l'instruction `#show` de la forme `p/n`, avec `p` le nom de l'atome associé à `n` arguments :

```
23 a(X) ← b(X).
24 b(1).
25 b(2).
26 #show a/1.
```

Seuls les atomes `a` seront affichés en résultat. Ici, on obtient : `a(1). a(2).`

Optimisation Avec les instructions d'optimisation, la question de savoir si un ensemble d'atomes est un ensemble-réponse est étendue à celle de déterminer s'il s'agit d'un ensemble-réponse optimal. Cette extension est exprimée grâce à des *contraintes faibles* dont la forme peut être comprise de façon similaire à celle des contraintes d'intégrité (présentées en section 4.2.2), à la différence qu'on y associe l'ensemble des termes qu'on souhaite minimiser avec un coût. La sémantique d'un programme avec des contraintes faibles est intuitive : un ensemble-réponse est optimal si son coût obtenu est minimal parmi tous les ensemble-réponses du programme donné (voir [Gebser et al., 2019] en section *Optimization*).

Résolution heuristique Il est possible d'incorporer des heuristiques dans la résolution ASP, directement à partir du programme logique ou via la ligne de commande, afin de calculer avec ces heuristiques un sous-ensemble de modèles stables. Cette opportunité offerte par clingo est exploitée dans le cadre de l'inférence de modèles de réseaux d'interactions biologiques pour augmenter la variabilité entre les solutions successives, afin

d'augmenter la diversité lors d'une énumération partielle. Cette notion est exploitée dans les applications présentées dans le chapitre 5.

4.3 Encodage de l'inférence de modèles en ASP

Dans cette section je vais détailler l'encodage ASP de l'inférence de réseaux booléens à partir de contraintes sur leur graphe d'interactions et sur leur dynamique.

Les contraintes sur la dynamique portent sur les configurations compatibles avec des observations afin de vérifier qu'elles respectent les propriétés souhaitées. Une observation `X` est spécifiée par un ensemble de prédicats `obs(X,N,V)`, où `N` et `V` désignent le composant et sa valeur booléenne observée. Les valeurs booléennes sont encodées par -1 pour faux et 1 pour vrai. Une configuration `X` l'est quant à elle par un ensemble de prédicats `cfg(X,N,V)`. Si le nœud `N` a été observé, `V` est égal à la valeur observée ; sinon, sa valeur est choisie :

```
1 cfg(X,N,V) ← obs(X,N,V).
2 1 {cfg(X,N,(-1;1))} 1 ← obs(X,_,_), node(N), not obs(X,N,_).
```

4.3.1 Domaine des réseaux booléens

L'encodage ASP des réseaux booléens localement monotones compatibles avec un graphe d'interactions se heurte à deux difficultés. Premièrement, deux solutions distinctes doivent correspondre à deux réseaux booléens f et f' non-équivalents, c'est-à-dire qu'il doit exister $x \in \mathbb{B}^n$ tel que $f(x) \neq f'(x)$. Cela nécessite de garantir que les solutions correspondent avec les représentations canoniques des réseaux booléens. Deuxièmement, dans le pire des cas, la taille de la spécification d'une fonction booléenne est exponentielle selon le nombre de ses variables. Par conséquent, l'encodage doit permettre de préciser une limite sur la taille de la fonction booléenne, idéalement sans limiter le nombre de variables.

Je représente les fonctions booléennes composant un réseau booléen sous leur forme normale disjonctive (DNF), tel que présenté en section 2.1.1. Afin d'avoir des DNF minimales, deux clauses distinctes ne doivent pas avoir de relation de sous-ensemble. Concrètement en ASP, les DNF ont été encodées comme des listes de clauses en attribuant un indice à chaque clause. À partir de là, la canonicité est assurée en imposant un ordre total entre les clauses. Le nombre maximal de clauses pour une DNF avec d variables est $\binom{d}{\lfloor d/2 \rfloor}$, et notre encodage permet de spécifier un nombre plus petit afin de restreindre l'ensemble des DNF à considérer, sans limiter le nombre de variables à considérer.

Globalement, l'encodage des fonctions booléennes canoniques avec d variables génère $O(ndk^2)$ atomes et $O(nd^2k^2)$ règles, avec k la limite supérieure fixée sur le nombre de clauses par fonction locale, le maximum étant $\binom{d}{\lfloor d/2 \rfloor}$. Avec cette valeur maximale, le nombre de solutions correspond au nombre de fonctions booléennes

monotones distinctes, c'est-à-dire au nombre de Dedekind [Kleitman, 1969]. Les nombres de Dedekind (actuellement connus jusqu'à $d = 8$ [Wiedemann, 1991]) sont présentés dans le tableau 4.1.

Nombre de variables booléennes	Nombre de Dedekind
0	2
1	3
2	6
3	20
4	168
5	7581
6	7828354
7	2414682040998
8	56130437228687557907788

Tableau 4.1 – Nombre de fonctions booléennes monotones possibles selon le nombre de variables booléennes.

Lorsque le k spécifié est inférieur au maximum, des fonctions booléennes ne sont pas capturées par l'encodage. Les contraintes sur la canonicité sont nécessaires pour obtenir une énumération efficace des solutions, mais elles peuvent de ce fait être relâchées si on souhaite uniquement vérifier l'existence d'au moins une solution. Faire ce choix réduit le nombre de prédicats et de règles à $O(ndk)$.

4.3.1.1 Encodage des fonctions booléennes

J'ai utilisé un prédicat `clause(N,C,L,S)` pour spécifier que le littéral `L` avec le signe `S` est inclus dans la `C`-ème clause de la DNF de $f_{\texttt{N}}$. Par exemple, la DNF à deux clauses $f_a(x) = (\neg x_a \wedge x_b) \vee x_c$ est codée par les trois prédicats suivants :

- `clause(a,1,a,-1)` avec dans l'ordre des arguments :
  - `a` pour indiquer qu'on décrit f_a,
  - `1` pour indiquer qu'on décrit la $1^{ère}$ clause de la DNF,
  - `a` pour indiquer que cette clause contient l'atome x_a,
  - `-1` pour indiquer que c'est la négation de cet atome.
- `clause(a,1,b,1)`
- `clause(a,2,c,1)`

Le graphe d'interactions Le domaine des arguments `N`, `L`, et `S` est entièrement déterminé par le graphe d'interactions d'entrée (V, E_+, E_-) ; `C` allant de 1 à k compris. Le graphe d'interactions est codé avec des prédicats `node/1` tels que `node(`i`)` si et seulement si $i \in V$, et des prédicats `in/3` tels que `in(`j`,`i`,1)` si et seulement si $(j, i) \in E_+$, ainsi que `in(`j`,`i`,-1)` si et seulement si $(j, i) \in E_-$. La limite sur le nombre de clauses est fixée par `maxC(N,`k`)` :

```
3 {clause(N,1..C,L,S): in(L,N,S), maxC(N,C), node(N), node(L)}.
```

La monotonie locale est assurée en refusant, au sein de la DNF de chaque composant `N`, qu'un littéral puisse apparaître à la fois avec le signe positif et négatif.

```
4 ← clause(N,_,L,S), clause(N,_,L,-S).
```

Les DNF sans clause donnent lieu à des fonctions constantes, spécifiées par le prédicat `constant/2` :

```
5 1 {constant(N,(-1;1))} 1 ← node(N), not clause(N,_,_,_).
```

Canonicité La canonicité est obtenue en garantissant que les clauses sont ordonnées par taille puis par ordre lexicographique, et sans relation de sous-ensemble. L'ordre par taille est garanti par les contraintes d'intégrité présentées en l.6-8. La l.6 garantit que les identifiants des clauses augmentent continuellement à partir de 1.

```
6 ← clause(N,C,_,_), not clause(N,C-1,_,_), C > 1.
7 size(N,C,X) ← clause(N,C,_,_), X = #count{L,S: clause(N,C,L,S)}.
8 ← size(N,C1,X1), size(N,C2,X2), X1 < X2, C1 > C2.
```

L'ordre lexicographique entre clauses de même taille est appliqué comme suit en l.9-11, avec `clausediff(N,C1,C2,L)` indiquant que `L` est présent dans la `C1`-ème clause mais pas dans la `C2`-ème ; et `mindiff(N,C1,C2,L)` indiquant que `L` est le plus petit littéral tel que `clausediff(N,C1,C2,L)`.

```
9 ← size(N,C1,X), size(N,C2,X), C1 > C2, mindiff(N,C1,C2,L1), mindiff(N,C2,C1,L2), L1 < L2.
10 clausediff(N,C1,C2,L) ← clause(N,C1,L,_), not clause(N,C2,L,_), clause(N,C2,_,_), C1 != C2.
11 mindiff(N,C1,C2,L) ← clausediff(N,C1,C2,L), L <= L' : clausediff(N,C1,C2,L'); clause(N,C1,L',_), C1 != C2.
```

Enfin, l'absence de relation de sous-ensemble entre clauses d'une DNF est garantie par la contrainte d'intégrité suivante :

```
12 ← size(N,C1,X1), size(N,C2,X2), X1 <= X2, clause(N,C2,L,S): clause(N,C1,L,S); C1 != C2.
```

4.3.2 Évaluation des fonctions booléennes

Des règles génériques ont été définies pour évaluer les fonctions booléennes sur les hypercubes (cf. définition 2.2.5). Un hypercube est spécifié de manière similaire aux configurations, avec les prédicats `mcfg(H,N,V)` dans lesquels `V` vaut -1 ou 1, mais avec potentiellement à la fois `mcfg(`h`,`i`,-1)` et `mcfg(`h`,`i`,1)` afin d'indiquer que le composant i est libre dans l'hypercube h ($h_i = *$). L'encodage des contraintes dynamiques prend soin d'instancier les `mcfg/3` connexes.

Les règles d'évaluation des fonctions booléennes garantissent la création d'un prédicat `eval(`h`,`i`,1)` (resp. `eval(`h`,`i`,-1)`) si et seulement s'il existe une configuration $x \in c(h)$ telle que $f_i(x)$ est vrai (resp. false). Une clause est évaluée à faux lorsque l'un de ses littéraux est évalué à faux (l.13), et à vrai si tous ses littéraux sont évalués à vrais (l.14). Ensuite, pour l'évaluation de la DNF :

- soit la fonction est une constante et son évaluation suit la valeur de la constante (l.17),
- soit l'une de ses clauses est vraie et par conséquent elle est évaluée à vrai (l.15),
- soit toutes ses clauses sont fausses et par conséquent elle est évaluée à faux (l.16).

L'encodage prend en compte la possibilité de forcer l'expression ou le silence de certains composants afin de modéliser des perturbations telles que des mutations. Le modèle de prédicat utilisé est `clamped(H,N,V)`.

```
13 eval(H,N,C,-1) ← clause(N,C,L,-V), mcfg(H,L,V), not clamped(H,N,_).
14 eval(H,N,C,1) ← clause(N,C,_,_), mcfg(H,_,_), mcfg(H,L,V): clause(N,C,L,V), not clamped(X,N,_).
15 eval(H,N,1) ← eval(H,N,C,1); clause(N,C,_,_).
16 eval(H,N,-1) ← clause(N,_,_,_), mcfg(H,_,_), eval(H,N,C,-1): clause(N,C,_,_).
17 eval(H,N,V) ← constant(N,V), mcfg(H,_,_), not clamped(X,N,_).
18 eval(X,N,V) ← clamped(X,N,V).
```

Pour chaque hypercube, cet encodage génère $O(nk)$ prédicats et $O(ndk)$ règles.

Afin de faire correspondre la dynamique du réseau booléen avec le comportement des observations biologiques, j'ai distingué deux familles de contraintes : les contraintes *existentielles* vérifiant l'existence d'un comportement au sein de la dynamique du réseau booléen, et les contraintes *universelles* vérifiant l'exclusivité d'un comportement au sein de la dynamique.

4.3.3 Propriétés existentielles

4.3.3.1 Contrainte d'atteignabilité positive

Chaque $(m, m') \in$ PR est traduit en prédicat `reach(`m`,`m'`)`, spécifiant que la configuration x^m doit pouvoir atteindre la configuration $x^{m'}$. Étant donné la définition 3.1.4 de compatibilité entre un réseau booléen et une liste d'observations, les propriétés d'atteignabilité avec la sémantique MP peuvent être évaluées avec des hypercubes particuliers. La règle ci-dessous déclare un hypercube dédié à la contrainte d'atteignabilité positive, initialement égal à la configuration initiale.

```
19 mcfg((pr,X,Y),N,V) ← reach(X,Y), cfg(X,N,V).
```

Ensuite, l'hypercube doit être étendu pour satisfaire la propriété de *trap space* contraint telle que présentée en définition 2.2.6. Les extensions des hypercubes sont encodées avec les prédicats `ext(H,N,V)`, et leur application par la règle générique présentée à la ligne 20. Pour un composant `N`, on distingue deux règles pour l'extension de l'hypercube selon la valeur à laquelle `N` est évalué :

- Si l'évaluation de la fonction du composant `N` d'un hypercube aboutit à la valeur de `N` dans la configuration cible `Y`, alors l'hypercube est étendu pour inclure cette valeur (l.21).

- Si la fonction peut être évaluée à la valeur opposée de celle de la configuration cible, son inclusion dans l'hypercube est un choix (l.22).

```
20 mcfg(H,N,V) ← ext(H,N,V).
21 ext((pr,X,Y),N,V) ← reach(X,Y), eval((pr,X,Y),N,V), cfg(Y,N,V).
22 {ext((pr,X,Y),N,V)} ← reach(X,Y), eval((pr,X,Y),N,V), cfg(Y,N,-V).
```

L'hypercube qui en résulte est un *trap space* contraint sur L, avec L l'ensemble des composants pour lesquels les extensions créées par la ligne 22 ont été ignorées, à condition que la valeur opposée ne figure pas déjà dans la configuration initiale.

Au final, les deux propriétés que le *trap space* contraint doit vérifier (déf. 2.2.5) mènent aux règles suivantes :

l.23 : Rejeter les modèles pour lesquels la configuration cible n'est pas incluse dans l'hypercube.

l.24 : Rejeter les modèles pour lesquels un composant est libre dans l'hypercube (donc absent de L), mais dont la valeur cible ne peut pas être obtenue avec cette fonction au sein du *trap space* contraint sur L.

```
23 ← cfg(Y,N,V), not mcfg((pr,X,Y),N,V), reach(X,Y).
24 ← cfg(Y,N,V), not ext((pr,X,Y),N,V), ext((pr,X,Y),N,-V), reach(X,Y).
```

Étant donné les règles liées à `eval`, pour chaque prédicat `reach(X,Y)`, $O(nk)$ prédicats et $O(ndk)$ règles sont générés.

4.3.3.2 Contrainte d'atteignabilité négative

Chaque $(m, m') \in$ NR est traduit en prédicat `nonreach(`m`,`m'`)`, spécifiant qu'il est impossible d'atteindre la configuration $x^{m'}$ à partir de la configuration x^m. La propriété d'atteignabilité en *Most Permissive* dépend de l'existence d'un sous-ensemble de composants $L \subseteq \{1, \ldots, n\}$ tel que w le plus petit *trap space* contraint sur L qui contient la configuration initiale remplit les deux conditions suivantes :

1. w contient également les configurations cibles,
2. pour chaque composant i absent de L, il existe une configuration $z \in c(w)$ telle que $f_i(z) = y_i$.

Prouver l'absence d'atteignabilité exigerait que ces conditions ne soient vérifiées par aucun des sous-ensembles de composants L. Dans [Chatain et al., 2018], il a été démontré qu'il est suffisant de considérer au plus n sous-ensembles particuliers de composants L pour conclure à l'absence d'atteignabilité. Nous commençons donc par vérifier les conditions avec $L = \emptyset$, puis on ajoute successivement dans L les composants qui ne satisfont pas la seconde condition. Avec cette démarche, il est suffisant de vérifier la première condition dans l'ensemble L obtenu à la $n^{\text{ème}}$ itération.

Pour évaluer l'inatteignabilité de la configuration y à partir de x, notre encodage génère n hypercubes, initialement égaux à x (l.25-26). Ensuite, les prédicats `locked(X,Y,I+1,N)` spécifient que le composant `N` est dans l'itération `I+1`

de L. Un tel prédicat doit être vrai si `N` ne vérifie pas la seconde condition à l'itération `I` (l.27), ou s'il est déjà dans L à la précédente itération (l.28). L'extension de l'hypercube à l'itération `I` est ensuite contrainte par les composants dans L (l.29). Au final, s'il existe un composant `N` tel que $y_{\mathtt{N}}$ n'est pas l'hypercube de la dernière itération, le prédicat `nr(`x`,`y`)` est vrai, indiquant l'absence d'atteignabilité (l.30). Un modèle est rejeté si un tel prédicat ne peut pas être prouvé comme étant vrai (l.31).

```
25 iter(1..K) ← nbnode(K).
26 mcfg((nr,X,Y,I),N,V) ← nonreach(X,Y), cfg(X,N,V), iter(I).
27 locked(X,Y,I+1,N) ← cfg(X,N,V), cfg(Y,N,V), not ext((nr,X,Y,I),N,V), ext((nr,X,Y,I),N,-V), iter(I+1).
28 locked(X,Y,I+1,N) ← locked(X,Y,I,N), iter(I+1).
29 ext((nr,X,Y,I),N,V) ← not locked(X,Y,I,N), eval((nr,X,Y,I),N,V).
30 nr(X,Y) ← not mcfg((nr,X,Y,K),N,V), nbnode(K), cfg(Y,N,V), nonreach(X,Y).
31 ← not nr(X,Y), nonreach(X,Y).
```

Étant donné les règles liées à `eval`, pour chaque prédicat `nonreach(X,Y)`, cet encodage génère $O(n^2k)$ prédicats et $O(n^2dk)$ règles.

4.3.3.3 Contraintes de stabilité

Comme indiqué dans 4.1.1, on considère deux propriétés différentes en lien avec les attracteurs d'un réseau booléen f :

- propriété de **points fixes**, pour laquelle les configurations spécifiées doivent être des points fixes de f afin de correspondre à des marqueurs de stabilité totale ;
- propriété de **confinement**, où les configurations spécifiées doivent appartenir à des *trap spaces* contraints sur un sous-ensemble de composants afin de correspondre à des marqueurs de stabilité partielle. Tous les attracteurs au sein de ces *trap spaces* ont leurs configurations compatibles avec ces marqueurs de stabilité.

Étant donné les règles liées à `eval`, l'encodage de chacune de ces propriétés génère $O(nk)$ prédicats et $O(ndk)$ règles.

Points fixes Chaque $m \in$ FP est traduit en un prédicat `is_fp(`m`)`, spécifiant que la configuration x^m est un point fixe de f. La contrainte est assurée par le rejet des modèles pour lesquels l'évaluation aboutit à une valeur opposée pour au moins l'un des composants :

```
32 mcfg(X,N,V) ← is_fp(X), cfg(X,N,V).
33 ← is_fp(X), cfg(X,N,V), eval(X,N,-V).
```

Confinements Chaque $(m,i) \in$ TP est traduit en un prédicat `is_tp(`m`,`i`)`, spécifiant que le plus petit *trap space* t contenant la configuration x^m doit avoir le composant i fixé, c'est-à-dire $t_i \neq *$. L'initialisation et l'extension du plus

petit *trap space* contenant x sont obtenues avec les règles présentées aux lignes 34-35. Le modèle est rejeté si le *trap space* obtenu a n'importe quel composant libre spécifié comme contraint (l.36).

```
34 mcfg((ts,X),N,V) ← cfg(X,N,V), is_tp(X,_).
35 mcfg((ts,X),N,V) ← eval((ts,X),N,V).
36 ← is_tp(X,N), cfg(X,N,V), mcfg((ts,X),N,-V).
```

4.3.4 Propriétés universelles

Une propriété universelle implique une formule booléenne quantifiée avec 2 niveaux de quantificateurs, de la forme $\exists x \forall y : P(x, y)$. En ASP, afin d'explorer un ensemble de valeurs et de vérifier le respect d'une propriété pour chacune, les travaux de [Eiter and Gottlob, 1995] ont introduit une technique dite de saturation dont le principe repose à la fois sur une règle disjonctive (cf. section 4.2.5) et sur une saturation sur le terme soumis à la disjonction. La stratégie est de saturer l'ensemble-réponse avec les prédicats soumis à la disjonction afin d'exploiter le critère de minimalité d'ASP.

Par exemple, étant donné la règle suivante qui déterminerait pour chaque acteur (défini par un prédicat `actor/1`) sa présence (1) ou son absence (-1) :

```
37 scene(A,-1); scene(A,1) ← actor(A).
```

La saturation consisterait ici à forcer la présence de l'autre prédicat de la disjonction, si l'ensemble d'acteurs remplit la condition requise ϕ. Pour l'illustration ligne 38, il y a une unique règle saturante, mais on peut avoir besoin de distinguer différents cas de figure et pour cela cumuler plusieurs règles saturantes.

```
38 scene(A,-V) ← scene(A,V), φ.
```

Quelque soit l'ensemble d'acteurs considéré, celui-ci doit respecter la condition ϕ. Elle est donc requise pour que l'interprétation soit un ensemble-réponse :

```
39 ← not φ.
```

De ce fait, si une interprétation I ne remplit pas la condition ϕ, elle est éliminée (l.39). Mais si elle remplit la condition ϕ, l'ensemble-réponse est saturé (l.38). Or le critère de minimalité oblige le solveur à explorer tous les sous-ensembles de prédicats pour vérifier qu'aucun d'entre eux n'est une solution plus petite. Dans cet exemple, on parcourrait ainsi tous les ensembles d'acteurs possibles.

4.3.4.1 Contrainte universelle sur les points fixes (atteignables)

J'ai exploité cette technique de saturation pour assurer des contraintes universelles sur les points fixes ou sur des points fixes atteignables à partir d'une configuration donnée, afin d'exiger des modèles dont les seuls points fixes

possibles (ou seuls points fixes atteignables depuis une configuration d'intérêt) soient ceux compatibles avec des observations spécifiées avec des marqueurs de stabilité totale. C'est une contrainte qui était attendue notamment pour décrire les phénotypes qui sont atteignables selon différentes conditions. Par exemple, deux phénotypes stables (A et B) ont été observés en l'absence de mutation, alors qu'en présence de la mutation 1, ce furent les phénotypes B et C. Cette contrainte garantit que tous les points fixes (atteignables) sont compatibles avec un ensemble donné d'observations.

Principe de l'implémentation Il est nécessaire de parcourir l'ensemble des configurations possibles afin de vérifier la compatibilité des points fixes avec les propriétés souhaitées. Pour cela, on laisse le solveur déduire une configuration z qui va pouvoir correspondre à n'importe quelle configuration possible en associant à chaque nœud au moins une valeur d'état :

```
40 cfg(z,N,-1) ; cfg(z,N,1) ← node(N).
```

En effet, le modèle de prédicat `cfg(X,N,V)` assigne la valeur `V` au littéral `N` dans la configuration `X`. Par la règle en ligne 40, un ensemble de valeurs de nœuds est constitué pour définir une configuration z, avec le prédicat `cfg(z,N,_)` soumis à la sémantique de minimalité de sous-ensemble. Pour respecter la propriété désirée, chaque configuration z doit soit ne pas être un point fixe ($f(z) \neq z$), soit être compatible avec les observations souhaitées, c'est-à-dire avoir les mêmes états de composants que ceux spécifiés dans les prédicats dédiés à ces observations, soit ne pas être atteignable depuis une configuration d'intérêt dans le cas où on limite aux stabilités observées depuis une observation spécifiée. Si l'une de ces conditions est remplie, un prédicat `valid` est déduit (l.42 et 43).

Cas où z n'est pas un point fixe Une configuration n'est pas un point fixe si au moins l'un de ses composants peut changer d'état :

```
41 mcfg(z,N,V) ← cfg(z,N,V).
42 valid ← cfg(z,N,V) ; eval(z,N,-V).
```

Ainsi, le modèle de prédicat `mcfg(X,N,V)` créé en ligne 41 entraîne l'évaluation de la configuration `X` étant donné les règles booléennes du réseau. Puisque les valeurs atteignables sont stockées dans le prédicat `eval(X,N,V)` (cf. section 4.3.2), la condition est vérifiée en l.42 par le fait que `z` n'est pas un point fixe s'il est possible d'évaluer un composant `N` à la valeur opposée de celle qu'il a dans `z`.

Cas où z est un point fixe Dans le cas où `z` est un point fixe, il doit avoir les mêmes états de composants que ceux spécifiés dans une observation `X` marquée par le prédicat `is_universal_fp(X)`, qui est exprimé par la règle ASP suivante :

```
43 valid ← cfg(z,N,V):obs(X,N,V); is_universal_fp(X).
```

Cas où z n'est pas atteignable On permet de restreindre la propriété universelle aux points fixes qui sont atteignables à partir d'une configuration initiale donnée. Ceci est spécifié par le modèle de prédicat `is_universal_fp(X,S)`, où `S` désigne la configuration initiale, et `X` une observation, comme utilisé ci-dessus. En combinant de tels prédicats, on peut spécifier des ensembles de phénotypes atteignables. Pour cette variante, l'encodage contient une troisième façon de déduire `valid` : la non-atteinte de la configuration z à partir de `S`.

```
44 mcfg((ufp,S),N,V) ← cfg(S,N,V), is_universal_fp(X,S).
45 mcfg((ufp,S),N,V) ← eval((ufp,S),N,V).
46 valid ← cfg(z,N,V), not mcfg((ufp,S),N,V).
```

La stratégie consiste à calculer le *trap space* minimal contenant `S` (l.44-45) puis à vérifier que la configuration `z` n'en fait pas partie (l.46).

Saturation Le prédicat `valid` entraîne la saturation de la configuration z :

```
47 cfg(z,N,-V) ← cfg(z,N,V), valid.
```

Par conséquent, quand `valid` est déduit, l'ensemble-réponse contient toutes les valeurs possibles de composants pour z. Du fait de la sémantique de minimalité des sous-ensembles-réponses, le solveur est forcé d'assurer qu'il n'existe pas de sous-ensemble-réponse. Et la seule façon de trouver un tel ensemble-réponse plus petit serait de trouver un z à partir duquel `valid` ne pourrait être déduit, c'est-à-dire qui serait un contre-exemple de la propriété universelle. Or dans ce cas, la contrainte en l.48 élimine l'ensemble-réponse :

```
48 ← not valid.
```

En complément, notre implémentation permet de spécifier des mutations qui peuvent être combinées avec l'atteignabilité et avec des contraintes universelles sur les points fixes atteignables. Ça permet de considérer des observations sur le devenir de cellules soumises à différentes perturbations.

4.3.4.2 Limitation

Dans le chapitre 3, j'ai défini des contraintes universelles sur les attracteurs qui ne sont pas abordées par l'implémentation. La raison principale est que c'est un problème de complexité supérieure à ce qu'on peut exprimer en ASP (formule booléenne avec 3 niveaux de quantificateurs).

Résumé du chapitre 4

La méthode d'inférence de modèles dynamiques repose sur la stratégie qui consiste à décrire le problème d'inférence de réseaux booléens comme un problème de satisfiabilité que l'on peut résoudre avec des méthodes de satisfaction de contraintes. Nous l'avons implémentée sous la forme d'un programme logique qui contient à la fois la description du formalisme de modélisation dynamique (réseau booléen) et les données du processus biologique (mesures expérimentales, comportement observé, hypothèses).

Ce chapitre présente la formalisation de ce problème. Une mesure expérimentale est prise en compte sous la forme d'une association entre un composant et une valeur booléenne, et un ensemble de ces associations est appelé une observation. Le comportement biologique observé est ensuite décrit en apposant sur ces observations des propriétés dynamiques dont la méthode d'inférence va garantir le respect au sein des réseaux booléens inférés. Nous décrivons deux propriétés importantes, l'atteignabilité (positive) entre observations afin de modéliser l'évolution linéaire de mesures, et ce que nous avons appelé l'atteignabilité négative qui, en décrivant la non atteinte d'une observation à partir d'une autre, permet de modéliser des comportements plus complexes incluant des bifurcations. Les comportements stables observés peuvent être décrits sous la forme de points fixes au sein de la dynamique du modèle, mais nous avons également introduit la notion de *confinement* de composants qui permet de n'imposer la stabilité que sur un sous-ensemble des composants du système. L'ensemble des propriétés décrites jusqu'ici garantissent l'inférence de réseaux booléens compatibles avec des listes d'observations et des bifurcations ainsi que des marqueurs de stabilité totale et partielle, sous leurs définitions existentielles telles que présentées au chapitre 3. La compatibilité avec les marqueurs de stabilité totale (atteignables) peut être étendue selon la définition universelle afin de modéliser les états stables d'un système selon différentes conditions, contraignant les points fixes sur l'ensemble de la dynamique du modèle ou sur une sous-partie délimitée par une atteignabilité.

La mise en œuvre de notre méthode d'inférence repose sur la programmation par ensemble-réponse (*Answer-Set Programming* - ASP). Ce chapitre rappelle les bases de ce langage de programmation logique, en introduisant la syntaxe nécessaire à la compréhension des contraintes implémentées dans le cadre de la méthode d'inférence, mais également la sémantique des *modèles stables* sur laquelle repose ASP. Ces bases permettent d'introduire également la *technique de saturation*, stratégie utilisée pour aborder des propriétés nécessitant une formulation booléenne quantifiée, grâce au couplage entre la propriété de minimalité engendrée par la sémantique des modèles stables et la possibilité en ASP d'exprimer des règles particulières dites disjonctives.

Enfin, ce chapitre présente une contribution majeure de ma thèse qui est le développement en ASP des contraintes logiques d'atteignabilité (positive et négative) et de stabilité correspondant aux propriétés décrites précédemment. Intégrées au programme logique, elles limitent l'inférence aux seuls réseaux booléens reproduisant les comportements décrits. Le principe et l'implémentation en ASP de chacune de ces contraintes sont décrits dans ce chapitre. Les contraintes existentielles ont été publiées dans [Chevalier et al., 2019], contribution présentée à la conférence ICTAI 2019, et l'existentielle dans [Chevalier et al., 2020], présentée à la conférence CMSB 2020.

Chapitre 5

BoNesis : présentation et applications

Sommaire

Par la création des contraintes logiques décrites au chapitre 4, j'ai participé au développement d'un outil de synthèse automatique de réseaux booléens, appelé BoNesis. Cet outil permet d'étendre la synthèse de modèles à des processus biologiques aux propriétés dynamiques complexes non prises en compte par les autres outils de modélisation. Il rend ainsi possible la modélisation de mécanismes régulateurs de l'évolution des cellules tels que ceux gouvernant les processus de différenciations cellulaires, qu'ils soient sains (embryogenèse, renouvellement cellulaire) ou pathologiques (cancer).

Le développement de BoNesis a permis plusieurs applications sur des données biologiques. Premièrement, j'ai participé à la modélisation d'une voie de signalisation impliquée dans la progression du cancer, en collaboration avec l'institut Curie. Le travail réalisé alors a permis d'illustrer la synthèse et la simulation d'ensembles de modèles afin de prendre en compte la variabilité des modèles possibles. Cette démarche de modélisation a été présentée dans l'article *"Synthesis and Simulation of Ensembles of Boolean Networks for Cell Fate Decision"* [Chevalier et al., 2020] publié à la conférence CMSB 2020. Une seconde application a concerné la différenciation des cellules du sang, appelée hématopoïèse. Cette modélisation m'a permis d'illustrer l'utilisation de BoNesis, d'une part, à partir de données de séquençage single-cell et, d'autre part, sans requérir en amont de la modélisation de connaissances expertes sur les interactions impliquées dans le comportement observé. J'y montre en effet comment BoNesis peut aider la délicate étape de la construction d'un *Prior Knowledge Network* (PKN) pertinent au regard du processus à modéliser. Un article présentant ce travail est en finalisation en vue d'être soumis à une revue scientifique en direction de la communauté bioinformatique.

Ce chapitre détaille les fonctionnalités et l'utilisation de BoNesis par une présentation de l'outil dans la première section, suivie de la description des deux applications introduites ci-dessus afin d'illustrer son utilisation.

5.1 BoNesis

BoNesis est un outil de synthèse automatique de modèles dynamiques sous la forme de réseaux booléens. Il a été conçu sur l'idée d'aborder cette modélisation comme un problème de satisfiabilité booléenne. Son principe est de décrire, sous la forme d'un problème logique à résoudre, la recherche d'un réseau booléen compatible avec un graphe d'interactions et avec des propriétés dynamiques données. Pour cela, BoNesis intègre des connaissances et

des observations au sein d'un même programme logique, afin que toute solution de ce programme soit un réseau booléen constitué d'interactions appartenant à l'ensemble de celles possibles étant donné les connaissances, et dont les propriétés dynamiques sont compatibles avec le comportement des observations.

Le programme logique est écrit en Answer-Set Programming présenté au chapitre 4. Il décrit le problème de synthèse de modèles par des prédicats et des contraintes non seulement les données biologiques (les connaissances sur les interactions, les observations et leur comportement), mais également le formalisme de la modélisation (le réseau booléen et le calcul de sa dynamique en sémantique Most Permissive). Les contraintes sont des conditions nécessaires et suffisantes qui garantissent que toute solution du problème est un réseau booléen compatible avec les données biologiques, c'est-à-dire, un réseau booléen inclus dans le domaine délimité par les connaissances et dont les propriétés dynamiques sont compatibles avec le comportement des observations. Les solutions sont obtenues grâce à un solveur par ensemble-réponse, *clasp* [Gebser et al., 2012], les modèles étant les ensemble-réponses satisfaisant le programme logique.

BoNesis est disponible sous forme d'un package python, développé par Loïc Paulevé (`https://github.com/bnediction/bonesis`), qui implémente les modélisations ASP que j'ai conçues et présentées dans le chapitre 4. Un tutoriel sur le dépôt github illustre son utilisation.

5.1.1 Données biologiques considérées

Avec BoNesis, toute solution est un réseau booléen compatible avec, d'une part, un état des connaissances sur les interactions entre composants biologiques et, d'autre part, des observations sur ces composants au cours du processus biologique qu'on souhaite modéliser.

L'état des connaissances est couramment représenté sous la forme d'un graphe d'interactions tel que présenté en figure 5.1 et défini dans la section 2.1.4. Appelé *Prior Knowledge Network* (PKN), il est usuellement constitué de gènes et peut inclure d'autres composants tels que des protéines, des médicaments et des phénotypes. Pour BoNesis, le graphe d'interactions est décrit par une liste d'interactions, chacune étant constituée d'une paire de composants associée à un signe afin d'indiquer l'effet activateur (1) ou inhibiteur (-1) du premier composant sur le second.

Les observations apportent quant à elles des informations sur l'évolution de composants. Ce sont par exemple des relevés d'expression de gènes issus de séquençages transcriptomiques, l'administration de médicaments et des constats de phénotypes, associés à différents points de temps d'une expérience ou à différentes cellules. Pour BoNesis, une observation est un ensemble d'informations binaires tel que nous l'avons défini en 3.1.1. Entre le recueil des données et leur utilisation par BoNesis, une étape de binarisation est donc nécessaire. Une matrice typique collectant des données utilisables par BoNesis est montrée en figure 5.2, chaque ligne correspondant à une

observation et chaque colonne à un composant observé. La valeur NA y indique une valeur booléenne indéterminée.

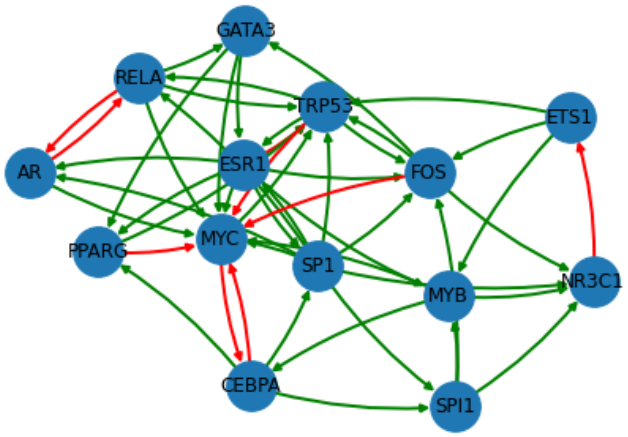


FIGURE 5.1 – ***Prior Knowledge Network*** : interactions à considérer pour le modèle. Flèche verte : effet activateur, flèche rouge : effet inhibiteur.

	AR	ETS1	FOS	MYB	RELA	SPI1
obs1	0	NA	0	0	1	1
obs2	NA	0	NA	0	0	1
obs3	0	0	1	0	0	NA
obs4	1	1	1	1	1	0
obs5	1	1	0	NA	1	1
obs6	1	0	0	0	0	0

FIGURE 5.2 – **Observations** : matrice de données, où les colonnes sont les composants et les lignes les différentes observations. Les valeurs possibles sont 0, 1 et NA (valeur indéterminée).

5.1.2 Fonctionnalités de BoNesis

À partir de la recherche d'un réseau booléen compatible avec les données, deux utilisations complémentaires de BoNesis sont possibles :

énumération des modèles : rechercher les autres réseaux booléens compatibles afin d'énumérer tout ou partie des modèles ;

sélection de composants : rechercher un réseau booléen maximisant un certain critère dans le but d'identifier des composants que l'on peut mettre de côté. Cette utilisation permet de raffiner le *Prior Knowledge Network* en amont de l'énumération des modèles.

Ces deux utilisations de BoNesis sont complémentaires pour la création de modèles, tout particulièrement pour une synthèse automatique à partir d'un large domaine de connaissances issu de bases de données d'interactions. En effet, une étape préliminaire importante à la construction d'un modèle dynamique d'un réseau d'interactions consiste à déterminer le *Prior Knowledge Network*, c'est-à-dire, quels vont être les composants et interactions qui pourront être inclus dans le modèle. Le cœur du circuit de régulation d'un processus étant généralement gouverné par une dizaine de composants, la construction d'un PKN est une tâche complexe qui requiert une grande expertise du processus à modéliser. C'est une étape délicate et experte car, d'une part, inclure des composants et interactions ne jouant aucun rôle dans le processus observé fait exploser la combinatoire et entrave l'analyse des modèles et, d'autre part, manquer des composants importants peut rendre impossible la construction d'un modèle reproduisant les observations. Avec BoNesis, nous proposons un moyen d'aider à la conception d'un PKN pertinent au regard des données, en étant capable de confronter un large graphe d'interactions, tel qu'on peut l'extraire d'une base de données d'interactions comme DoRothEA [Garcia-Alonso et al., 2019] ou Signor [Licata et al., 2019], aux

observations recueillies sur le processus à modéliser. Cette utilisation offre la possibilité de considérer un maximum de connaissances pour mener la recherche de modèles, en utilisant BoNesis pour mettre de côté des composants déterminés comme non pertinents pour l'explication des propriétés dynamiques des observations. Je détaille dans la sous-section suivante la stratégie utilisée pour réaliser ce raffinement du PKN.

L'utilisation de BoNesis peut être étendue pour considérer davantage de besoins. Par exemple, il permettrait également de considérer d'autres critères d'optimisation, par exemple minimiser l'utilisation de certaines influences peu fiables au sein du PKN lorsque cette information est disponible.

5.1.2.1 Sélection de composants

Lorsqu'on considère un *Prior Knowledge Network* de grande taille, les observations recueillies sur le processus à modéliser n'apportent des informations que sur un petit nombre des composants du PKN, parmi lesquels seule une poignée est réellement impliquée dans le processus. Or une solution du programme logique est un réseau booléen inclus dans le PKN et compatible avec les observations. Ça implique que si le PKN contient des composants qui n'entrent pas dans la régulation du phénomène observé, de nombreuses fonctions différentes peuvent être attribuées à ces composants sans impacter la compatibilité du réseau booléen avec les observations. Ces composants sans importance pour le phénomène observé augmentent alors fortement le nombre de solutions sans apporter d'information. D'autre part, si aucune combinaison de fonctions ne permet de reproduire les données car il manque des composants et des interactions clés du processus, aucun modèle ne pourra être trouvé. Pour répondre à cette problématique de la construction d'un graphe d'interactions pertinent étant donné les observations recueillies, nous proposons une stratégie utilisant BoNesis.

Nous fixons un critère d'optimisation, afin que le solveur cherche un réseau booléen compatible qui maximise le nombre de composants d'un type particulier, appelés *constantes fortes*. Une constante forte est un composant auquel est attribué une fonction constante, et dont la valeur reste constante pour reproduire les observations au sein de la dynamique du réseau booléen. Ainsi, au sein d'un réseau booléen compatible avec les données biologiques, un nœud A est une constante forte si et seulement si $f(A) = v$ avec $v \in \{-1, 1\}$ et qu'au sein de la dynamique du réseau booléen il est possible de reproduire les données avec le nœud A toujours égal à v (toutes les configurations x associées aux observations ont $x_A = v$). Autrement dit, c'est un composant ayant ni activateur ni inhibiteur et dont la valeur peut rester inchangée sans empêcher le réseau booléen d'être compatible avec les observations. Ces constantes fortes peuvent être supprimées du domaine sans impacter la compatibilité avec le comportement des données. C'est pourquoi le programme logique créé par BoNesis permet de rechercher un réseau booléen compatible avec des données ayant le maximum de constantes fortes. En général, il peut y a voir plusieurs ensembles maximaux de constantes fortes. Une fois le domaine des interactions réduit aux composants qui ne sont pas des constantes fortes, on peut se limiter à la composante fortement connexe (*CFC*) maximale de ce graphe, CFC particulièrement

intéressante pour se concentrer sur les interactions qui régulent le processus observé.

Étant donné la taille potentiellement conséquente du PKN à raffiner, des contraintes sur la dynamique du réseau booléen peuvent s'avérer trop coûteuses en ressources pour permettre au solveur la recherche d'une solution dans un délai raisonnable. Cela peut être le cas de l'accessibilité négative et de l'universalité sur les points fixes, contraintes pouvant être alors incluses dans un second temps après une première sélection résultant de la prise en compte des autres contraintes.

5.1.2.2 *Diversité* dans l'énumération des modèles

Il est fréquent que les observations ne soient pas assez contraignantes au regard du PKN considéré ce qui amène à un très grand nombre de réseaux booléens compatibles. En effet, le faible nombre de conditions d'observations ne permet souvent pas de discriminer les différentes combinaisons logiques (ET/OU) entre les régulateurs d'un composant ; or ce nombre de combinaisons est exponentiel suivant le nombre de régulateurs. Le nombre de modèles possibles peut donc exploser à cause de quelques composants. Pour autant, d'autres composants peuvent avoir des fonctions identiques dans l'ensemble des modèles énumérés et des motifs d'interactions peuvent être hautement partagés, révélant des éléments clés de la régulation de ce processus.

C'est pourquoi ne considérer qu'un sous-ensemble des modèles possibles peut être suffisant pour avoir un bon aperçu des modèles possibles. Mais pour cela, il est essentiel d'avoir un sous-ensemble de modèles pertinents au regard de la diversité des réseaux booléens dont la dynamique est compatible avec les données considérées. Or les solveurs ASP explorent les solutions au gré de légères variations. Ainsi, une énumération partielle a de grandes chances de donner un ensemble de solutions successives fortement similaires, par exemple, distinctes sur la fonction d'un seul composant. Inspiré par [Razzaq et al., 2018], BoNesis emploie des heuristiques du solveur clingo [Gebser et al., 2014] pour le pousser vers des solutions éloignées. Pour cela, à chaque solution, un sous-ensemble d'affectations de variables est sélectionné et il est demandé au solveur de les éviter dans les itérations suivantes. En appliquant cette heuristique, le solveur ne bénéficie plus d'une recherche de proche en proche ce qui ralentit l'énumération des modèles, mais les solutions successives sont plus variées qu'avec l'énumération standard, augmentant la diversité de l'ensemble de modèles obtenus avec une énumération partielle des solutions.

5.1.3 Comportements biologiques modélisables

Les contraintes décrites au chapitre 4 rendent possible la garantie de la compatibilité entre un réseau booléen et des propriétés dynamiques des données biologiques. En combinant ces contraintes, il est possible d'obtenir les réseaux booléens compatibles avec des comportements complexes tels que celui de données de différenciation.

5.1.3.1 Description d'une information d'ordre entre observations

À la base du suivi d'un processus biologique, il y a l'information de son évolution. Les données collectées pour observer cette évolution du processus s'organisent sous la forme de listes d'observations (séries temporelles, pseudo-temps, couples d'observations avant et après une perturbation expérimentale...). Un modèle reproduisant l'évolution du processus est donc un modèle compatible avec la (ou les) liste(s) d'observations sur ce processus.

Afin de garantir cette compatibilité définie en 3.1.1.2, BoNesis intègre la contrainte d'atteignabilité positive (décrite en 4.3.3.1) qui garantit l'existence d'un chemin entre deux observations. L'obtention de modèles reproduisant l'évolution d'un processus observé est donc assurée en définissant une contrainte d'atteignabilité positive à chaque étape des listes d'observations collectées.

La figure 5.3 illustre la description de l'ordre liant les observations entre elles par l'ajout d'une contrainte d'atteignabilité positive d'une configuration compatible avec l'observation A à une configuration compatible avec l'observation B. Il existera donc, au sein de la dynamique du modèle, un chemin d'une configuration ayant $g_1 = 0, g_2 = 0, g_3 = 0$ à une configuration ayant $g_1 = 1, g_2 = 0, g_3 = 0$. Deuxièmement, il doit exister un chemin de cette seconde configuration à une configuration compatible avec l'observation C ($g_1 = 1, g_2 = 1, g_3 = 1$) étant donné la seconde contrainte d'atteignabilité.

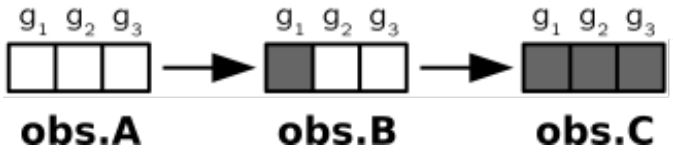


FIGURE 5.3 – Exemple de deux contraintes d'atteignabilité positive entre observations : de A vers B et de B vers C. Elles décrivent ainsi l'ordre entre les observations A, B et C : au sein de la dynamique du réseau booléen il doit exister un chemin entre configurations compatibles avec A, B et C, permettant d'aller de A à C en passant par B.

5.1.3.2 Description d'une information de stabilité sur une observation

Il est très fréquent de souhaiter reproduire via un modèle un comportement stable observé afin de modéliser un phénotype ou plus largement un état cellulaire stable. À l'échelle de l'ensemble de composants considérés pour une modélisation, ces états peuvent être décrits par une stabilité de l'ensemble ou d'un sous-ensemble de ces composants.

Afin de garantir la compatibilité définie en 3.1.11 entre un réseau booléen et ces informations de stabilité (qu'elle soit partielle ou totale), BoNesis intègre la contrainte de confinement (décrite en 4.3.3.3) ainsi que la contrainte de point fixe qui est spécifique à la stabilité totale (décrite en 4.3.3.3). L'obtention de modèles reproduisant l'information de stabilité associée à une observation est assurée en définissant l'une ou l'autre de ces contraintes suivant le besoin.

Confinement La contrainte de confinement garantit qu'au sein de la dynamique du réseau booléen il existe une configuration confinée sur l'ensemble des composants donné. Elle permet de prendre en compte une information de stabilité possiblement partielle, c'est-à-dire limitée à un sous-ensemble des composants utilisés pour décrire le système, afin de modéliser, par exemple, la stabilité connue d'un gène marqueur d'un phénotype (gène dont l'expression est caractéristique d'un type cellulaire).

La figure 5.4 illustre l'information d'une stabilité associée à l'observation K qui concerne le composant g_3. En posant la contrainte de confinement, on a la garantie qu'au sein du réseau booléen, il existe une configuration compatible avec l'observation K qui est confinée sur g_3 - autrement dit, une configuration à partir de laquelle seules des configurations avec g_3 actif sont atteignables.

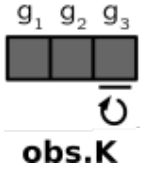


FIGURE 5.4 – Exemple d'une observation décrite comme ayant un composant stable. Un réseau booléen compatible avec ce marqueur de stabilité possède dans sa dynamique une configuration compatible avec l'observation K qui est confinée sur g_3.

Point fixe La contrainte de point fixe assure que, pour chaque observation désignée comme un marqueur de stabilité totale, il existe dans la dynamique du modèle un point fixe compatible avec l'observation. Il est ainsi tout à fait possible de créer des modèles à partir de données d'expression sur les états stables du système (usuellement appelées *steady-state expression data*) en appliquant la contrainte de point fixe sur les états stables observés. Un modèle de ce comportement est alors un réseau booléen dont la dynamique contient au moins un point fixe compatible avec chacun des états stables. Il est tout à fait possible de combiner ces informations de stabilité avec d'autres informations telles que celles d'ordres entre observations afin, par exemple, de décrire la stabilité d'un phénotype concluant une liste d'observations.

La figure 5.5 illustre l'information d'une stabilité totale de deux observations M et C, information prise en compte par l'ajout de deux contraintes de points fixes. Il doit donc exister dans la dynamique du réseau booléen un point fixe compatible avec l'observation M ($g_1 = 1, g_2 = 0, g_3 = 0$) et un point fixe compatible avec l'observation C ($g_1 = 1, g_2 = 1, g_3 = 1$).

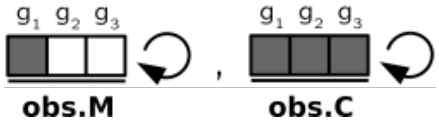


FIGURE 5.5 – Exemple de deux observations décrites comme états stables. La dynamique d'un réseau booléen compatible avec cet ensemble de marqueurs de stabilité totale inclut au moins un point fixe compatible avec l'observation M et un point fixe compatible avec l'observation C.

Point fixe universel BoNesis intègre une contrainte de point fixe dite universelle, qui permet d'assurer qu'au sein de l'ensemble ou d'une sous-partie du réseau booléen il existe non seulement des points fixes compatibles avec chacun des états stables observés, mais qu'il n'existe aucun point fixe incompatible. Cette contrainte correspond aux définitions de compatibilité 3.1.15 et 3.1.17. Grâce à elle, il est possible, par exemple, de décrire des données de mutation ou de perturbation, données au sein desquelles sont observées les évolutions de cellules soumises à différentes conditions expérimentales. En effet, pour modéliser ce type de comportement observé, il faut spécifier que l'ensemble des états stables observés est différent selon les conditions, avec certains états atteignables uniquement depuis certaines des conditions. Dans la dynamique du modèle, la contrainte universelle permet alors de garantir que, à partir de chaque condition expérimentale, seuls des points fixes compatibles avec les états stables spécifiés sont atteignables.

Expression ou silence forcé d'un composant Les observations réalisées au cours d'expérimentations peuvent être concernées par des actions permanentes sur des composants telles que des perturbations forçant l'expression ou le silence de gènes. Afin de prendre en compte cette information, BoNesis permet de renseigner les perturbations en lien avec une liste d'observations. Au sein du réseau booléen, cela se concrétise par une fonction constante pour le composant forcé pour la vérification des propriétés dynamiques liées à cette liste d'observations.

La figure 5.6 illustre cette information, avec une première liste d'observations soumises au silence forcé du composant g_1, une seconde liste d'observations soumises à l'expression forcée de g_2, et une troisième correspondant à la condition expérimentale sans perturbation.

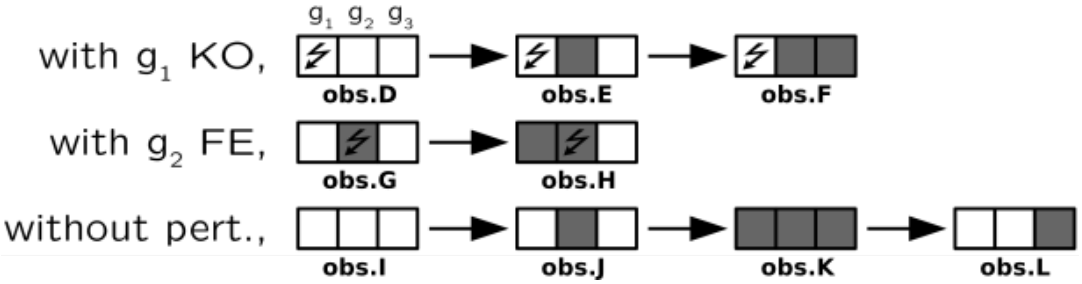


FIGURE 5.6 – Exemple de trois listes d'observations, dont deux sont associées à des perturbations : silence forcé de g_1 pour la première, expression forcée de g_2 pour la seconde. Un réseau booléen est compatible avec la première (resp. seconde) liste d'observations perturbées s'il est compatible avec la liste d'observations étant donné $g_1 = 0$ (resp. $g_2 = 1$).

5.1.3.3 Description d'une information de bifurcation entre listes d'observations

Il existe un besoin important de créer des modèles reproduisant la divergence de comportements cellulaires, c'est-à-dire l'évolution différente de cellules issues d'une même population cellulaire. En effet, ce comportement est caractéristique de nombreuses données recueillies en biologie : inhérentes au processus étudié (différenciations cellulaires) ou issues des conditions expérimentales (cellules soumises à différentes perturbations).

J'ai décrit ce phénomène de divergence en définissant au chapitre 3 un point de bifurcation et ses voies de différenciation (définition 3.1.6). Afin de garantir la compatibilité entre un réseau booléen et l'impossibilité de passer d'une voie de différenciation à une autre, BoNesis intègre la contrainte d'atteignabilité négative (décrite en 4.3.3.2) qui permet d'assurer l'absence de chemin d'une configuration à une autre, typiquement entre configurations appartenant aux voies de différenciation issues du même point de bifurcation.

La figure 5.7 illustre cette information entre deux observations : la contrainte de non-atteignabilité définie entre les observations B et J garantit qu'au sein du réseau booléen il existe une configuration compatible avec B à partir de laquelle il est impossible d'atteindre une configuration compatible avec J.

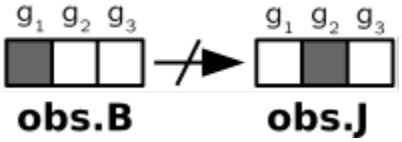


FIGURE 5.7 – Exemple de deux observations impliquées dans une contrainte d'atteignabilité négative. Dans un réseau booléen compatible avec cette atteignabilité négative il est impossible d'atteindre une configuration J depuis une configuration compatible avec B.

5.1.3.4 Description d'un comportement complexe

De très nombreux comportements biologiques observés peuvent être décrits par une combinaison d'informations d'ordres, de stabilités et de bifurcations entre les observations réalisées.

La différenciation cellulaire est un exemple de ce type de comportement qui nécessite une modélisation prenant en compte l'ensemble des propriétés décrites au sein d'un arbre de différenciation (cf définition 3.1.22). La figure 5.8 illustre ce type de données, avec deux voies de différenciation possibles à partir de l'observation A $g_1 = 0, g_2 = 0, g_3 = 0$. Ce comportement peut être décrit grâce aux contraintes d'atteignabilité pour les listes d'observations des voies de différenciation, auxquelles s'ajoute une contrainte de non-atteignabilité de l'observation B à l'observation D pour assurer l'absence de chemin de la première vers la seconde voie, ainsi qu'une contrainte de stabilité du gène 3 sur l'observation B et d'une contrainte de point fixe sur l'observation D.

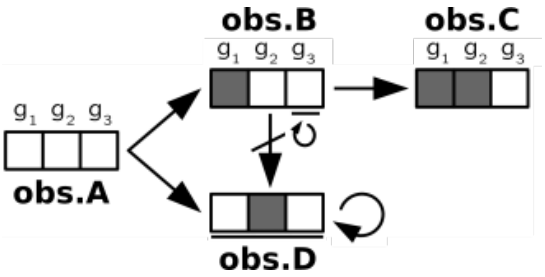


FIGURE 5.8 – Exemple de contraintes définies pour décrire le comportement d'observations recueillies au cours d'un processus de différenciation cellulaire.

5.2 Modélisation de la régulation du destin cellulaire dans la progression du cancer

La construction de modèles de réseaux biologiques à partir de connaissances préalables et de données expérimentales conduit souvent à une multitude de modèles possibles. En effet, malgré les énormes progrès des technologies expérimentales, les observations des processus biologiques restent très partielles, que ce soit en termes de résolution temporelle, de nombre d'entités observées, de synchronisation entre les points de mesure ou de variété des conditions expérimentales. Cette caractéristique, combinée à la complexité des interactions entre les composants biologiques, rend le problème de synthèse de modèles très souvent largement sous-spécifié, ce qui conduit à un nombre de modèles admissibles trop importants pour qu'ils puissent être tous synthétisés et étudiés. Dans ce contexte, la conception d'un unique modèle dépend fréquemment de choix arbitraires, ce qui peut conduire à des biais importants dans les prédictions ultérieures.

À partir du travail réalisé pour la synthèse de modèles et intégré dans BoNesis, j'ai participé à une démarche de modélisation et de simulation permettant désormais de prendre en compte la variabilité des modèles d'un processus biologique. Pour cela, la méthodologie appliquée consiste dans un premier temps à synthétiser des ensembles de modèles booléens satisfiant les propriétés dynamiques du comportement observé grâce à BoNesis. Puis, à partir de cet ensemble de modèles, l'outil de simulations stochastiques MaBoSS [Stoll et al., 2017] a été étendu afin de permettre des prédictions à l'échelle de l'ensemble des modèles obtenus et non plus d'un unique modèle. Cette démarche de modélisation d'ensemble permet d'améliorer la robustesse des prévisions en tenant compte de la variabilité et de l'incertitude potentielles du modèle.

Afin d'illustrer cette démarche, je présente dans cette section une application réalisée en collaboration avec l'équipe U900 de l'institut Curie, sur un modèle booléen précédemment publié d'une voie de signalisation régulant le destin cellulaire dans la progression du cancer [Cohen et al., 2015]. L'objectif a été de comparer les prédictions sur le modèle publié avec celles obtenues en considérant un ensemble de modèles. Comme dans l'étude originale, nous avons évalué le changement de phénotypes atteignables causé par l'interaction entre plusieurs mutations. Afin de décrire le contexte de l'application, je présente tout d'abord le modèle de base et l'analyse de sa dynamique via des simulations. Je décris ensuite les conditions de réalisation de la synthèse de modèles et la stratégie mise en place pour obtenir un résultat de simulations sur l'ensemble des modèles obtenus.

5.2.1 Modèle de base

Nous illustrons notre approche de modélisation d'ensemble sur un modèle publié de décision du destin cellulaire menant soit aux premiers événements de la métastase, soit à la mort cellulaire par apoptose [Cohen et al., 2015].

Cette décision est affectée par les déclencheurs initiaux, tels que des dommages à l'ADN ou des signaux de changements environnementaux (respectivement, les composants *DNAdamage* et *ECMicroenv* dans la figure 5.9), mais également par l'activité de certains gènes ou protéines participant au processus. La voie de signalisation impliquée fait intervenir les protéines TGFbeta, WNT, bêta-caténine, p53 et ses homologues, ainsi que certains micro-ARN et des facteurs de transcription de la transition épithélio-mésenchymateuse (*EMT*). La figure 5.9 montre le graphe d'interactions du réseau booléen publié.

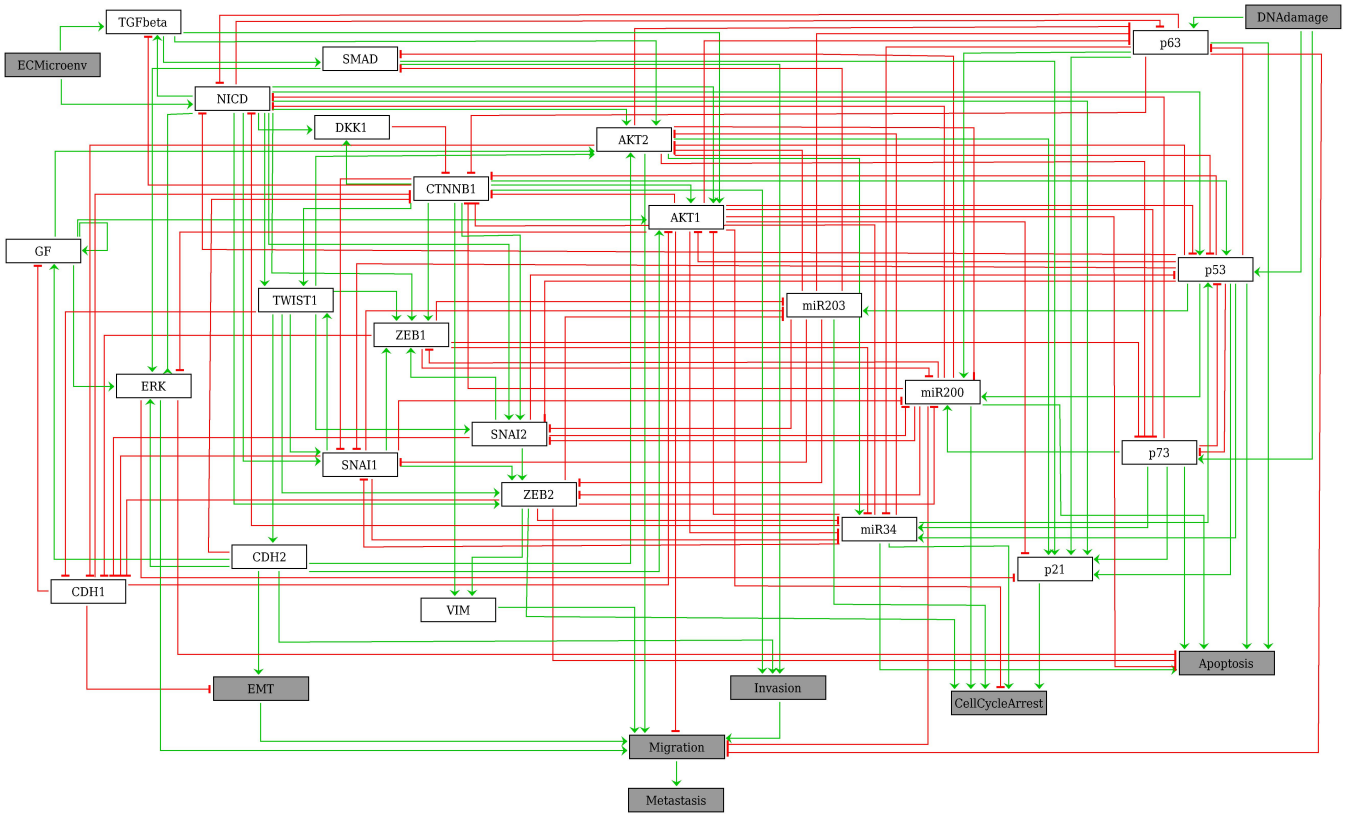


FIGURE 5.9 – Graphe d'interactions du modèle de Cohen reliant 32 composants avec 159 arcs, où les arcs activateurs sont en vert et les inhibiteurs en rouge.

Les fonctions de ce réseau booléen, que nous appelons "modèle de Cohen", ont été conçues manuellement afin que les simulations correspondent aux données expérimentales relatives à l'observation de phénotypes stables selon différentes mutations isolées. La publication initiale a ensuite exploré l'effet de combinaisons de ces mutations, en particulier quelle synergie conduit à des phénotypes métastatiques.

5.2.1.1 Analyse du modèle de Cohen

Nous avons reproduit une partie de l'analyse de [Cohen et al., 2015] sur le modèle original de Cohen en calculant les attracteurs atteignables à partir de 4 conditions initiales possibles, présentées dans le tableau 5.1. Au sein de ces conditions initiales, tous les nœuds sont inactifs à l'exception, d'une part, des miRNAs qui sont actifs (miR34, miR200, miR203) et, d'autre part, des composants modélisant les dommages à l'ADN (*DNAdamage*) et les signaux

environnementaux (*ECMicroenv*) qui sont laissés libres.

composant	**actif (1) ou inactif (0)**
DNAdamage	$\{0, 1\}$
ECMicroenv	$\{0, 1\}$
miR34/200/203	1
autres	0

Tableau 5.1 – 4 conditions initiales possibles selon les 4 couples de valeurs possibles pour *DNAdamage* et *ECMicroenv*.

Nous avons tout d'abord considéré la condition dite *sauvage* (*WT*) sans aucune mutation. Ce modèle comporte 9 points fixes qui correspondent chacun à un phénotype physiologique identifié (selon les ensembles de valeurs présentés dans le tableau 5.2) : apoptose, EMT (transition épithélio-mésenchymateuse), métastase (équivalent à migration) et état homéostatique (HS). En n'appliquant pas de mutation, le résultat des simulations (en proportion des phénotypes atteints quelle que soit la condition initiale) est représenté en figure 5.10(a).

	Apoptose	**EMT**	**Métastase**	**HS**
Apoptosis	1	0	0	0
CellCycleArrest	1	1	1	0
EMT	0	1	1	0
Invasion	0	0	1	0
Metastasis	0	0	1	0
Migration	0	0	1	0

Tableau 5.2 – Valeurs de nœuds identifiant les 4 phénotypes physiologiques principaux.

Nous avons ensuite considéré une condition dite *double-mutant*, qui correspond à la fois à la perte de fonction (LoF) de p53 et au gain de fonction (GoF) de NICD. Ce modèle présente, quant à lui, un unique point fixe correspondant au phénotype métastase. En appliquant ces deux mutations (par fixation de p53 à 0 et NICD à 1), toutes les simulations aboutissent donc au même phénotype (figure 5.10(b)).

5.2.2 Analyse des ensembles de modèles

5.2.2.1 BoNesis : synthèse de deux ensembles de 1 000 modèles

Afin de montrer l'impact des fonctions booléennes alternatives, des ensembles de réseaux booléens ont été synthétisés avec BoNesis. Au sein de ces ensembles, tous les modèles partagent exactement le même graphe d'interactions que le modèle de Cohen et respectent les propriétés dynamiques voulues pour décrire le comportement observé. Ainsi, les modèles reproduisent l'existence et l'absence de propriétés d'atteignabilité entre les conditions initiales et les phénotypes physiologiques identifiés, l'existence de points fixes correspondant à ces phénotypes, ainsi que des propriétés universelles sur les points fixes et les points fixes atteignables, toutes ces propriétés pouvant être associées à la présence de mutations.

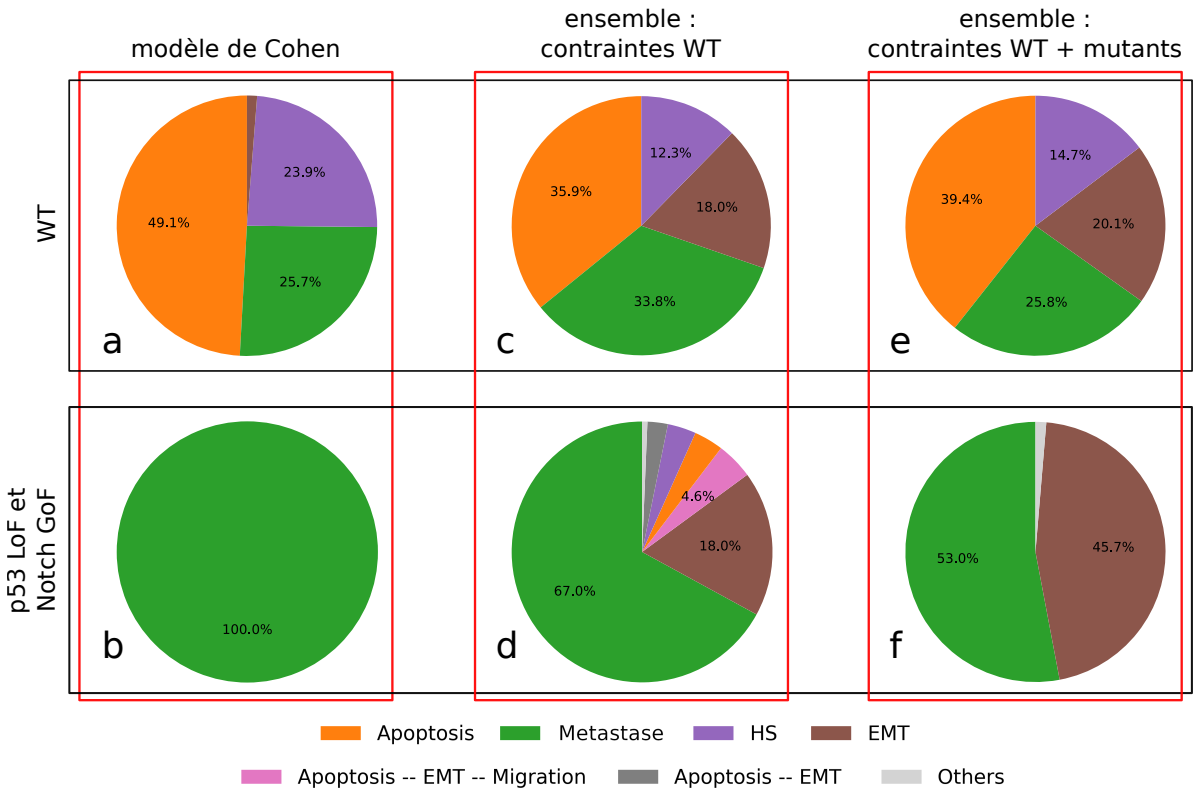


FIGURE 5.10 – Les proportions des phénotypes obtenus par simulations pour : (a,b) le modèle de Cohen, (c,d) l'ensemble obtenu à partir des contraintes WT, (e,f) l'ensemble obtenu à partir des contraintes WT et des mutants uniques (e,f). Les diagrammes a,c,e correspondent à la condition de type sauvage, tandis que b,d,f correspondent à la condition de double mutant p53 LoF/NICD GoF.

Concrètement, deux ensembles de chacun 1 000 réseaux booléens ont été synthétisés. Dans ces mille réseaux booléens, la présence d'attracteurs cycliques a été interdite. Le premier ensemble garantit uniquement le comportement de type sauvage (WT) ce qui signifie, d'une part, que tous les points fixes correspondent à l'un des quatre phénotypes physiologiques (contrainte universelle sur les points fixes) et, d'autre part, que chaque phénotype physiologique est accessible à partir d'au moins une des conditions initiales (contraintes d'atteignabilité positive). Le second ensemble garantit des propriétés supplémentaires :

- avec l'unique mutation p53 LoF, le comportement est le même qu'en WT ;
- avec l'unique mutation NICD GoF, seuls deux des phénotypes WT sont observés (EM et métastase) ainsi qu'un troisième phénotype correspondant à une situation intermédiaire, caractérisé par l'activation à la fois de l'arrêt du cycle cellulaire (*CellCycleArrest*), de l'EMT et de l'invasion (contraintes universelles sur les points fixes atteignables).

5.2.2.2 Simulation d'un ensemble de modèles

Afin de réaliser des simulations stochastiques sur des ensembles de réseaux booléens, une extension de l'outil de simulations stochastiques MaBoSS a été développé par Vincent Noël de l'équipe U900 à l'institut Curie (*Ensemble*

MaBoSS simulation présenté dans [Chevalier et al., 2020]). Le résultat d'une simulation basée sur un ensemble de modèles est une distribution multidimensionnelle, constituée d'autant de vecteurs de probabilités d'attracteurs qu'il y a de modèles dans l'ensemble. Cet ensemble de prédictions peut être agrégé en réalisant une moyenne de ces vecteurs, afin d'obtenir une prédiction au niveau de la population cellulaire qui tienne compte de son hétérogénéité. En outre, la variance de la distribution des probabilités peut être explorée en appliquant, par exemple, une méthode standard d'apprentissage automatique telle que l'analyse en composantes principales (ACP). Cette visualisation permet d'obtenir un aperçu de la diversité des modèles constituant l'ensemble.

Dans le cadre de l'application, la dynamique des deux ensembles obtenus a été explorée avec MaBoSS par Vincent Noël, afin de quantifier les attracteurs atteignables en considérant possiblement des perturbations du réseau. En conservant les mêmes paramètres que ceux appliqués pour le modèle de Cohen, des simulations stochastiques ont été réalisées avec les deux ensembles synthétisés, selon des taux d'activation et de désactivation uniformes. Alors que les comportements du WT sont similaires en considérant quelques différences dans les proportions des phénotypes (Fig. 5.10 (c,e)), le double-mutant met en évidence une différence notable entre le modèle unique et les ensembles de modèles. En effet, bien que la métastase reste le résultat le plus probable, les ensembles de modèles permettent d'observer plusieurs autres phénotypes possibles. Ce résultat suggère que, contrairement à l'analyse initiale du modèle unique, il existe certainement une variabilité dans l'efficacité de la double mutation à augmenter le potentiel de métastase. Il est d'ailleurs intéressant de noter que les contraintes supplémentaires sur la dynamique du deuxième ensemble de modèles, décrivant les comportements observés lors des mutations isolées de p53 et NICD, ne sont pas suffisamment restrictives pour garantir le comportement très drastique du modèle de Cohen.

5.2.2.3 Variabilité des probabilités des phénotypes

Afin d'étudier la composition des ensembles de modèles, nous avons souhaité analyser comment se répartissent les prédictions de phénotypes des différents modèles. Cependant, selon les résultats, on peut avoir à parcourir un nombre important de phénotypes identifiés, ce qui pose un problème de dimensionnalité. Pour cette raison, Vincent Noël a représenté les probabilités à l'aide d'une analyse en composantes principales, afin de visualiser la répartition des prédictions dans un nombre réduit de dimensions.

L'ACP a été réalisée sur les vecteurs de probabilités des modèles de l'ensemble obtenu avec, en plus des contraintes WT, les contraintes supplémentaires sur les mutations isolées de p53 et NICD. Le résultat de l'ACP selon les deux premières composantes principales est montré en figure 5.11. La première composante, qui représente 56% de la variance observée, montre une corrélation négative entre les phénotypes apoptotique et EMT. La deuxième composante, qui représente 24% de la variance observée, montre une corrélation négative entre EMT sans migration et EMT avec migration. La répartition des résultats des simulations réalisées sans mutation (WT) témoigne d'une diversité des prédictions, ce qui illustre une bonne performance de l'heuristique d'échantillonnage de modèles divers.

Quant aux résultats des simulations réalisées avec la double mutation p53 LoF et NICD GoF, ils sont concentrés vers les phénotypes EMT et/ou métastase (équivalent à migration), loin des phénotypes apoptotiques.

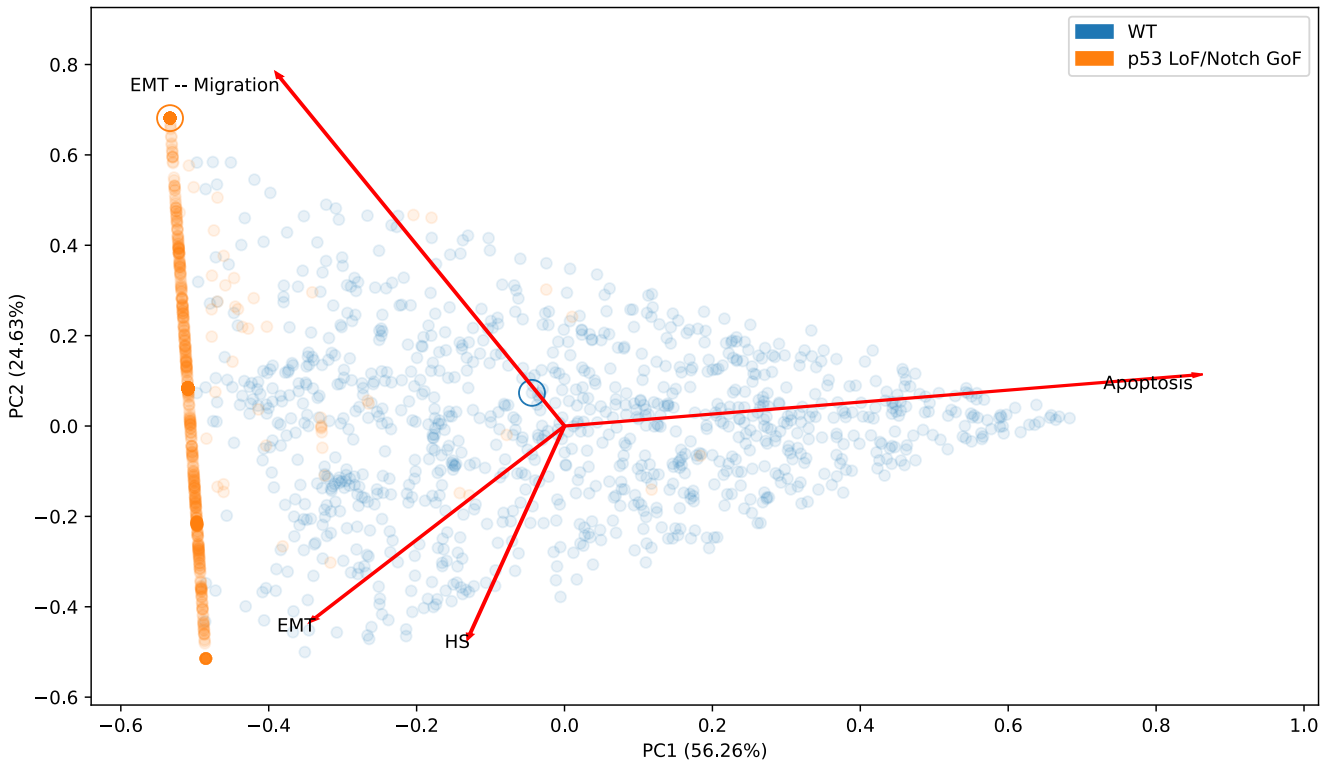


FIGURE 5.11 – Représentation en ACP de la distribution des états stables de chaque modèle parmi l'ensemble obtenu à partir des contraintes WT + mutations isolées. Chaque point représente le résultat de la simulation d'un seul modèle (les points bleus proviennent des simulations WT, les points orange des simulations p53 LoF/NICD GoF). Les cercles bleu (vers le centre) et orange (en haut à gauche) mettent en évidence la position de la simulation du modèle original de Cohen. La forme triangulaire de la distribution provient du fait que les probabilités des phénotypes sont situées dans le simplex à n dimensions.

La modélisation booléenne basée sur les ensembles apporte ici un nouvel éclairage sur la combinaison des mutations p53 et NICD, en montrant la variabilité de son effet amenant une cellule cancéreuse à métastaser.

5.3 Modélisation de la régulation de l'hématopoïèse

Le séquençage *single-cell* d'ARN (scRNA-seq) permet d'observer l'expression des gènes dans une population hétérogène de cellules en différenciation et de retracer l'évolution souvent complexe ayant abouti à cette hétérogénéité. Cette information est exploitable avec BoNesis afin de modéliser le phénomène de différenciation observé. Pour illustrer cette modélisation de trajectoires évolutives complexes à partir de données *single-cell*, j'ai modélisé les interactions géniques sous-tendant la différenciation des cellules du sang, appelée hématopoïèse, à partir d'un jeu de données de séquençage issu des travaux présentés dans [Nestorowa et al., 2016]. Dans ces travaux, un séquençage transcriptomique *single-cell* de cellules sanguines de souris a été réalisé afin de cartographier la diversité des expressions de gènes au cours de l'hématopoïèse. Le jeu de données utilisé est le décompte normalisé du comptage GSE81682 de Gene Expression Omnibus. Cette information normalisée est disponible sur `blood.stemcells.cam.ac.uk/data/normalisedCounts.txt.gz`.

La première partie de cette section est dédiée à la description de la préparation que j'ai effectuée sur ces données single-cell afin de pouvoir les utiliser pour la synthèse de modèles. Cette préparation est spécifique aux données single-cell et constitue une étape essentielle. Les choix qui y sont faits ont un impact important sur les modèles qui seront ensuite obtenus puisqu'ils déterminent les observations et les propriétés dynamiques que les modèles devront reproduire.

Dans la seconde partie de cette section j'illustre une contribution importante de l'outil BoNesis découlant de la stratégie utilisée pour la synthèse automatique de réseaux booléens : la possibilité de sélectionner, au sein d'un très grand réseau d'interactions tel que ceux obtenus auprès des bases de données publiques, un ensemble de composants et d'interactions pertinents au regard des observations du processus. Je montre ici comment, à partir des interactions extraites de la base de données DoRothEA, j'ai sélectionné le domaine des interactions à considérer au sein des modèles de l'hématopoïèse.

Je termine la section par la méthodologie suivie pour l'énumération des modèles compatibles avec ces données *single-cell* et des analyses réalisées sur les modèles obtenus.

5.3.1 Traitement des données single-cell

Les données *single-cell* donnent accès à l'hétérogénéité des cellules observées. En cela elles ne fournissent pas une information directement exploitable par BoNesis : le lien évolutif qui lie les cellules observées entre elles est déduit de l'hétérogénéité des observations grâce à des outils de reconstruction de trajectoire tel qu'introduit en 1.1.2.4.

5.3.1.1 Reconstruire la trajectoire de différenciation

J'ai choisi d'utiliser l'outil STREAM [Chen et al., 2019] afin de reconstruire la trajectoire de différenciation à partir du décompte normalisé du séquençage ARN des cellules hématopoïétique du jeu de données. Cet outil d'inférence de trajectoire est particulièrement adapté à ce contexte puisqu'il a été développé pour décrire les processus dynamiques dont la trajectoire comprend de multiples points de bifurcation, tels que la différenciation ou la réponse aux stimuli. Cet outil reconstruit les trajectoires et infère le pseudo-temps à partir de différents types de données *single-cell*. La méthode ne nécessite pas de connaissances préalables telles que les points de temps, la cellule de départ ou le nombre d'événements de bifurcation pour reconstruire les trajectoires, de ce fait elle peut être utilisée directement à partir de données *single-cell*. Cet outil permet également une visualisation de la densité des différents types de cellules le long de la trajectoire, visualisation pratique pour étudier l'évolution des sous-populations de cellules et les gènes déterminant le destin cellulaire le long des trajectoires de différenciation.

Avec les données utilisées, j'ai obtenu la trajectoire présentée en figure 5.12. La trajectoire a la forme d'un arbre comportant deux bifurcations, visibles en figure 5.12. La racine de l'arbre a été facilement identifiée en observant le type des cellules ordonnées le long de la trajectoire. En effet, l'une des extrémités de segments contient une très forte présence de cellules déterminées comme cellules souches hématopoïétiques (HSC), cellules ayant le plus grand potentiel de différenciation parmi les types cellulaires présents dans cette population.

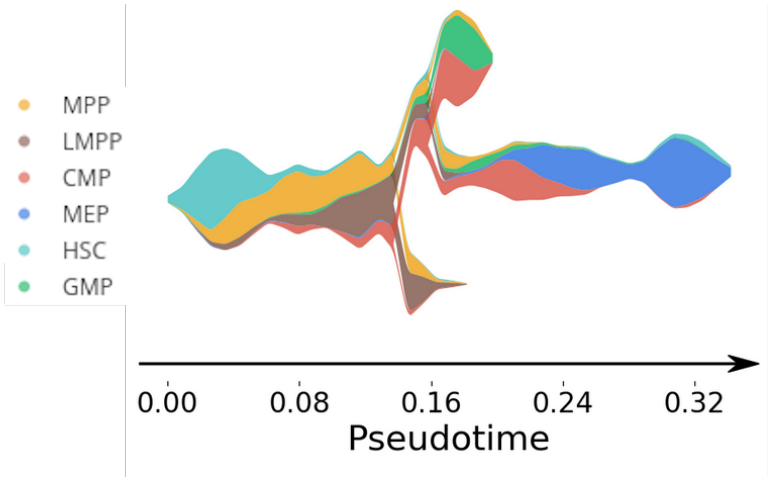


FIGURE 5.12 – Trajectoire de différenciation des cellules du sang (GSE81682) obtenue grâce à l'outil STREAM.

Le segment partant de la racine représente donc les premières étapes de la spécialisation cellulaire, avec une évolution de cellules souches hématopoïétiques (HSC) en progéniteurs multipotents (MPP). Le premier point de bifurcation marque l'évolution distincte de deux voies de différenciation : les progéniteurs myéloïdes communs (CMP) pour les segments supérieurs, les progéniteurs multipotents lymphoïdes (LMPP) pour le segment inférieur. Avec le deuxième point de bifurcation, on observe une différenciation entre les progéniteurs granulocytes-monocytes (GMP) et les progéniteurs mégacaryocytes-érythrocytes (MEP).

Parmi les 40 594 gènes ayant une expression non nulle dans au moins une cellule du jeu de données normalisé, STREAM a conservé 4 768 gènes dits *informatifs* pour déterminer la trajectoire.

De cette trajectoire peut être extrait un pseudo-temps et ainsi l'évolution de l'expression des gènes au cours de la différenciation, information que nous exploitons avec BoNesis. Pour cela, il est important de définir les observations entre lesquelles seront posées les contraintes qui vont décrire les propriétés dynamiques de la trajectoire de différenciation.

5.3.1.2 Créer les observations de la différenciation

Définition des observations

Parmi les biais inhérents aux données *single-cell*, il y a la sensibilité plus faible de la technique par rapport à un séquençage réalisé à l'échelle d'une population cellulaire. De ce fait, à l'échelle de la cellule, l'observation de transcrits (ARN) d'un gène est un signal clair de l'expression de ce gène au sein de la cellule, mais l'absence d'observation de transcrits d'un gène ne peut garantir la non expression de ce gène au sein de la cellule.

Afin d'atténuer l'impact de ce biais dû à la sensibilité, une stratégie est de réunir les mesures d'expression de plusieurs cellules afin de former une unique observation. En appliquant cette stratégie, j'ai décrit la différenciation des cellules sanguines par un ensemble d'observations aux étapes clés de la trajectoire reconstituée. Ainsi, les observations ont été construites en rassemblant les cellules se trouvant au voisinage de la racine, des points de bifurcation et des feuilles, tel qu'illustré par la figure 5.13. La taille de ces ensembles varie entre quelques dizaines à une centaine de cellules.

FIGURE 5.13 – Sélection d'un groupe de cellules autour des étapes clés de la trajectoire afin de créer les observations de la différenciation. La racine de la trajectoire a été déterminée grâce aux types des cellules le long de cette trajectoire : le nœud S1 concentre les cellules souches hématopoïétiques.

Binarisation des mesures d'expression

La binarisation des expressions de gènes a été réalisée grâce à la méthode PROFILE [Béal et al., 2019]. Cette

méthode détermine les seuils pour chaque gène en fonction de sa distribution de valeurs parmi les cellules, afin de conclure à une expression (valeur 1) ou non expression (valeur 0) du gène, avec la possibilité de ne pas conclure si la mesure est considérée comme non significative (valeur NA).

Afin de tirer profit du maximum d'information pour déterminer les seuils de binarisation, nous avons considéré la distribution de valeurs sur l'ensemble des cellules séquencées. Nous avons donc appliqué PROFILE sur la matrice de comptage contenant l'ensemble des cellules du jeu de données *single-cell* et non seulement celles correspondant aux étapes clés de la trajectoire. Nous avons ainsi obtenu une matrice de valeurs 0, 1 ou NA pour l'ensemble des cellules du jeu de données. J'ai ensuite déterminé les valeurs d'expression des gènes pour chaque observation de la manière suivante : l'expression d'un gène dans une observation correspond à la valeur majoritaire parmi les cellules de l'observation, la valeur attribuée pouvant être 0, 1 ou NA.

Suite à la binarisation, on constate des caractéristiques différentes de l'évolution des expressions selon les gènes observés. Parmi les 4 768 gènes retenus par STREAM, 1 519 ont des valeurs binaires différentes dans au moins deux des observations construites, par exemple 1 en S1 et 0 en S3. 1 369 d'entre eux sont binarisés dans toutes les observations, il n'y a donc pas d'indétermination sur leur valeur au cours de la trajectoire. 1 219 gènes ont des valeurs différentes entre les feuilles, donc entre les différentes spécialisations cellulaires possibles. Pour 905 gènes, la valeur binaire attribuée est la même à travers les observations ; cependant cela ne signifie pas que leur expression est stable dans la trajectoire puisqu'elle oscille peut-être en dehors des observations, créées aux extrémités des segments de la trajectoire.

5.3.1.3 Décrire les propriétés dynamiques de la différenciation

Le comportement d'un processus de différenciation a une forme d'arbre impliquant une combinaison de plusieurs propriétés dynamiques, tel que défini en 3.1.21. Premièrement, les arêtes de l'arbre impliquent la propriété d'atteignabilité de la racine aux feuilles : à partir d'une observation, il est possible d'atteindre les observations en aval de celle-ci dans l'arbre. Deuxièmement, les bifurcations de l'arbre impliquent la propriété de non atteignabilité entre arêtes en aval de la bifurcation : à partir des observations d'une voie issue de la bifurcation, il est impossible d'atteindre les observations d'une autre voie issue de la bifurcation. Troisièmement, les feuilles de l'arbre peuvent être interprétées comme des états stables que nous choisissons de décrire ainsi : à partir de l'observation d'une feuille il n'existe plus aucune évolution possible. Nous pouvons choisir de considérer les points fixes atteignables selon leur définition universelle, c'est-à-dire décrire l'ensemble des points fixes qui sont atteignables depuis des étapes clés de la différenciation.

Étant donné les 6 observations (S0 à S5) créées aux étapes clés de la trajectoire de différenciation, j'ai décrit les propriétés dynamiques de la trajectoire via les contraintes suivantes, où les 6 configurations associées doivent être distinctes :

- **Atteignabilités positives** pour imposer l'existence de trajectoires entre les observations, illustrées en figure 5.14. En considérant reach(X, Y) comme contrainte d'atteignabilité de X vers Y, la combinaison des 5 contraintes suivantes permet de reproduire l'évolution observée de la racine aux feuilles : reach$(S1, S0)$, reach$(S0, S2)$, reach$(S0, S3)$, reach$(S3, S4)$, reach$(S3, S5)$.

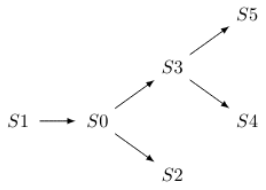


FIGURE 5.14 – Chaque flèche représente une contrainte d'atteignabilité positive d'une observation à une autre.

- **Points fixes** pour imposer aux feuilles de l'arbre de différenciation d'être des états stationnaires, illustrés en figure 5.15. En considérant fp(X) comme contrainte de point fixe sur X, les 3 contraintes suivantes imposent la stabilité des feuilles de l'arbre : fp$(S2)$, fp$(S4)$, fp$(S5)$.

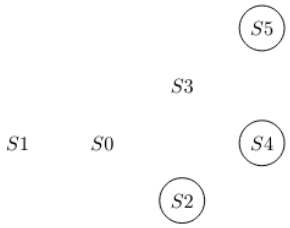


FIGURE 5.15 – Chaque cercle représente une contrainte de point fixe sur une observation.

- **Atteignabilité négative** pour imposer l'absence de chemin entre deux branches de différenciation, illustrée en figure 5.16. En considérant nonreach(X, Y) comme contrainte de non atteignabilité de X vers Y, la contrainte suivante garantit l'absence de chemin d'une voie à l'autre à l'issue de la première bifurcation de l'arbre : nonreach$(S3, S2)$.

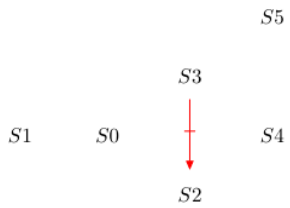


FIGURE 5.16 – La flèche représente une contrainte d'atteignabilité négative d'une observation à une autre. S2 étant un point fixe, il ne peut exister de trajectoire de S2 vers S3, tout comme entre S4 et S5.

- **Universalité des points fixes atteignables**, illustré en figure 5.17. En considérant univfp(X, Z) ajoutant Z à la liste des points fixes atteignables depuis X, les 2 contraintes universelles suivantes imposent que :
  - depuis S3, seuls S4 et S5 sont atteignables : univfp$(S3, S4)$ et univfp$(S3, S5)$
  - depuis S1, seuls S2, S4 et S5 sont atteignables : univfp$(S1, S2)$, univfp$(S1, S4)$ et univfp$(S1, S5)$

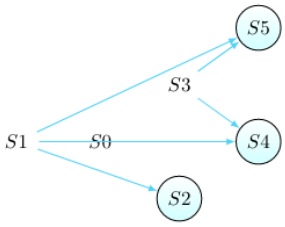


FIGURE 5.17 – Les flèches partant d'une même observation représentent l'ensemble des points fixes qu'il est possible d'atteindre depuis cette observation. Cette contrainte rend redondante celle d'atteignabilité négative précédemment décrite.

5.3.2 Obtention d'un domaine de connaissances en lien avec les observations

BoNesis peut être utilisé pour élaguer un domaine d'interactions en confrontant un grand réseau de connaissances préalables à des observations associées à des propriétés dynamiques. Ainsi, il permet de considérer en entrée l'ensemble des interactions provenant d'une base de données d'interactions, afin de ne conserver en sortie que les composants nécessaires pour reproduire les observations et les propriétés dynamiques associées. Cette stratégie a été suivie pour cette application sur l'hématopoïèse, avec un domaine de départ de plus de 5 000 composants directement obtenus à partir de la base de données publique d'interactions DoRothEA [Holland et al., 2020]. DoRothEA est un réseau de régulation génique contenant des interactions signées entre facteurs de transcription et gènes cibles. Ces interactions ont été collectées à partir de différentes sources pour l'homme et la souris, présentées sur l'illustration 5.18.

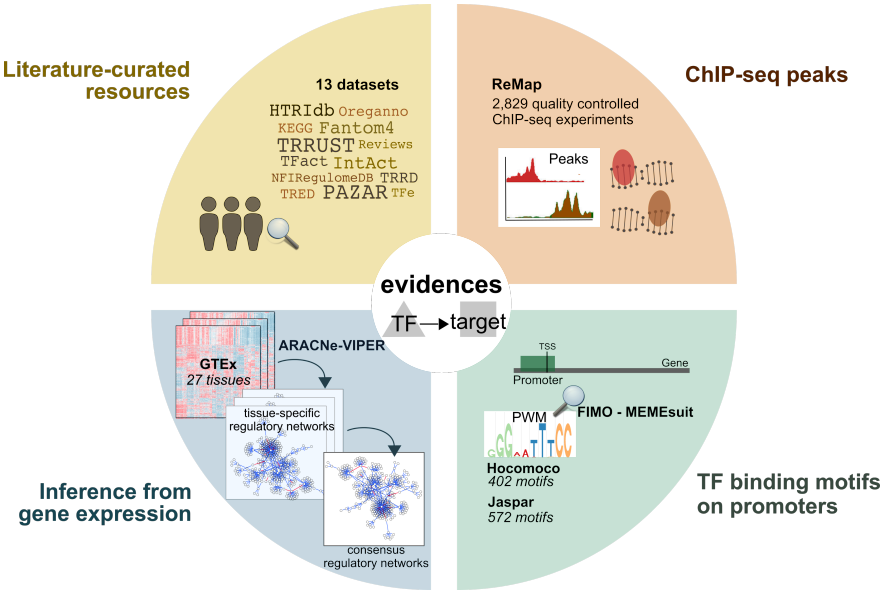


FIGURE 5.18 – Résumé des ressources et stratégies utilisées pour déduire les interactions TF-cible humaines, classées selon le niveau de preuve : ressources vérifiées manuellement (jaune), données expérimentales de liaison ChIP-seq (orange), prédiction des motifs de liaison TF basée sur les séquences des promoteurs de gènes (vert), ou inférence à partir des données GTEx (bleu). Cette figure est extraite de la publication [Garcia-Alonso et al., 2019]

5.3.2.1 Interactions considérées au sein de DoRothEA

Chaque interaction dans DoRothEA est associée à un indice de confiance allant de A (confiance la plus élevée) à E (confiance la plus faible), attribué en fonction de la fiabilité des données ayant servi à déterminer l'interaction. Ainsi, les interactions qui sont étayées par les 4 sources de données présentées en figure 5.18, vérifiées manuellement par des experts lors d'analyses spécifiques ou étayées par au moins deux ressources vérifiées sont considérées comme hautement fiables et sont étiquetées d'un A. Les niveaux B à D sont réservés aux interactions étayées par la littérature et/ou par des résultats ChIP-seq avec différents niveaux de preuves supplémentaires. Enfin, le niveau E est utilisé pour les interactions qui proviennent uniquement de prédictions informatiques.

Pour constituer notre domaine de départ, nous avons extrait de DoRothEA l'ensemble des interactions ayant un indice de confiance entre A et C. Cette extraction a conduit à un réseau de 12 895 arêtes reliant 5 186 composants. Deux catégories de composants nous intéressent dans ce graphe. D'une part, nous voulons conserver les composants dont l'évolution au cours du processus étudié a pu être observée au sein des données single-cell, puisque c'est cette évolution que nous allons modéliser. D'autre part, nous conservons les interactions impliquant des facteurs de transcription même en l'absence d'information sur l'évolution de ceux-ci au sein de notre jeu d'observations, puisqu'ils sont des composants clés de la régulation de l'expression des gènes. En limitant le graphe aux interactions entre facteurs de transcriptions ou entre un facteur de transcription et un gène observé, j'ai obtenu un graphe d'interactions de 2 777 arcs reliant 1 001 composants, parmi lesquels 849 ont leur expression observée dans l'ensemble de données *single-cell*.

5.3.2.2 BoNesis : sélection des composants pertinents au regard des observations du processus

Le domaine construit à partir de la base de données DoRothEA, contenant 1001 composants et 2777 arcs, a été confronté aux observations sur l'hématopoïèse afin de sélectionner les composants nécessaires à la reproduction du comportement observé. Ce comportement a été décrit par les contraintes existentielles suivantes présentées dans la section 5.3.1.3 : 5 accessibilités positives, une accessibilité négative et 3 points fixes, complétées par la contrainte universelle assurant que tous les points fixes atteignables depuis S3 sont compatibles avec S4 ou S5. La seconde contrainte universelle a été mise de côté pour cette première étape car, face à la très grande taille du domaine d'interactions, elle entraînait une augmentation trop importante du temps de résolution du problème ASP.

En considérant les interactions sélectionnées à partir de DoRothEA, les observations construites à partir des données *single-cell* et les contraintes décrivant le comportement observé, BoNesis a été utilisé pour construire un PKN pertinent au regard des informations sur le processus observé. La stratégie utilisée pour la sélection des composants est détaillée en 5.1.2.1. Cette étape a permis d'obtenir un PKN de 39 composants et 137 arcs présenté en figure 5.19.

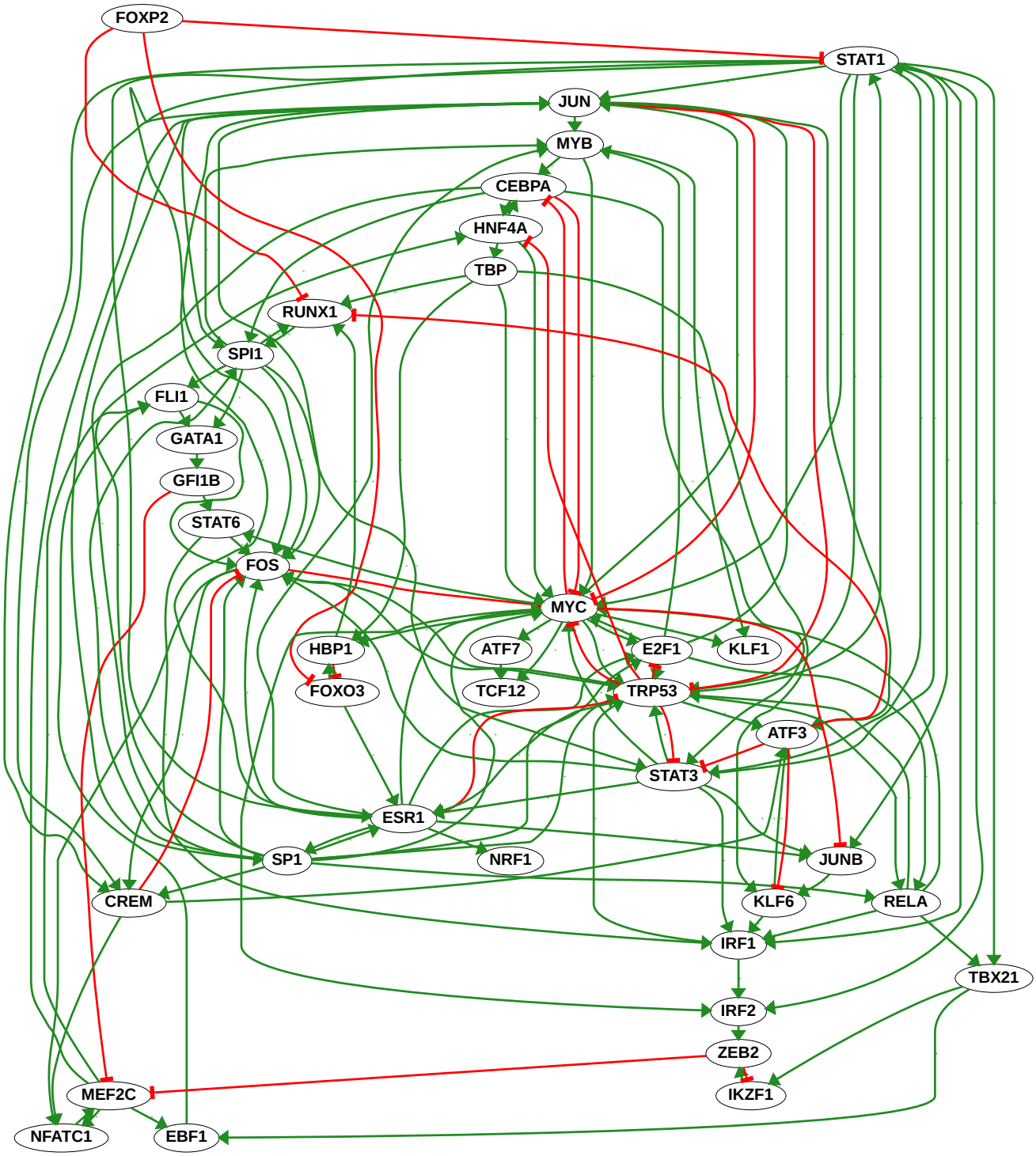


FIGURE 5.19 – PKN de 39 composants et 137 arcs obtenus par sélection de composants grâce à BoNesis, en confrontant les interactions extraites de DoRothEA avec les observations de l'hématopoïèse issues des données single-cell.

5.3.2.3 Analyse du domaine construit avec BoNesis

Comparaison avec les modèles existants sur l'hématopoïèse J'ai comparé notre domaine de gènes d'intérêt avec 3 modélisations de l'hématopoïèse de l'état de l'art. La première modélisation considérée provient des travaux de *Hamey et al.* [Hamey et al., 2017] qui ont reconstruit des réseaux de régulation des cellules souches sanguines à partir de profils d'expression single-cell. En considérant une partie des données de séquençage single-cell que nous avons nous-même utilisé, ils ont inféré deux modèles de réseau de régulation de gènes (usuellement appelé *gene regulatory network*) : l'un correspond à la voie de différenciation en MEP, l'autre en LMPP. Ces deux modèles considèrent au total 31 gènes. La seconde modélisation, issue des travaux de *Moignard et al.* [Moignard et al., 2015], est également un réseau de régulation de gènes construit à partir de profils d'expression single-cell. Il est constitué de 20 gènes. La troisième modélisation, de *Collombet et al.* [Collombet et al., 2017], est un réseau de régulation de gènes de 20 gènes construit à partir de données publiques et des résultats de plusieurs expériences (qPCR, RNA-seq, ChIP-seq).

Alors que ces modèles sont composés de 20 à 31 gènes, ils ne partagent que 2 gènes (ETS1 et IKZF1) et l'intersection des modèles deux-à-deux varie de 3 à 13 gènes comme le montre le diagramme en figure 5.20. Au total, 53 gènes distincts composent ces 3 modèles de la littérature. Parmi eux, 10 sont communs avec le PKN construit à l'aide de BoNesis à partir de la base de données DoRothEA et des données d'expression single-cell (CEBPA, EBF1, FLI1, GATA1, GFI1B, IKZF1, MEF2C, MYB, RUNX1, SPI1). En comparaison deux-à-deux, notre PKN a 6 gènes en commun avec chacun des modèles (couleurs ci-dessous en lien avec la figure 5.20) :

- *Hamey et al.* : FLI1, GATA1, GFI1B, IKZF1, MYB, RUNX1
- *Moignard et al.* : FLI1, GATA1, GFI1B, IKZF1, MYB, SPI1
- *Collombet et al.* : CEBPA, EBF1, IKZF1, MEF2C, RUNX1, SPI1

Analyse d'enrichissement fonctionnel J'ai réalisé une analyse d'enrichissement fonctionnel (souvent appelée *gene set enrichment analysis*) grâce à l'outil Metascape [Zhou et al., 2019], sur l'ensemble des 39 gènes constituant le PKN. Metascape est un portail web proposant plusieurs fonctionnalités pour l'annotation et l'analyse de listes de gènes, dont l'enrichissement fonctionnel qui permet d'identifier, étant donné un ensemble de gènes, les processus biologiques dans lesquels un maximum de gènes de cet ensemble sont impliqués. Pour cela, l'analyse d'enrichissement compare la liste de gènes d'entrée à des milliers d'ensembles de gènes, chacun de ces ensembles étant les gènes qu'on sait être impliqués dans un processus biologique spécifique. Cette connaissance est apportée par des ontologies telles que Gene Ontology [Ashburner et al., 2000], KEGG [Kanehisa, 2000] et Reactome [Jassal et al., 2019] qui sont, parmi d'autres, utilisées par Metascape pour l'analyse d'enrichissement. Les processus biologiques dont les gènes associés sont statistiquement sur-représentés dans la liste des gènes d'entrée sont mis en évidence par cette analyse. Cependant, l'interprétation du résultat est souvent assez complexe en raison non

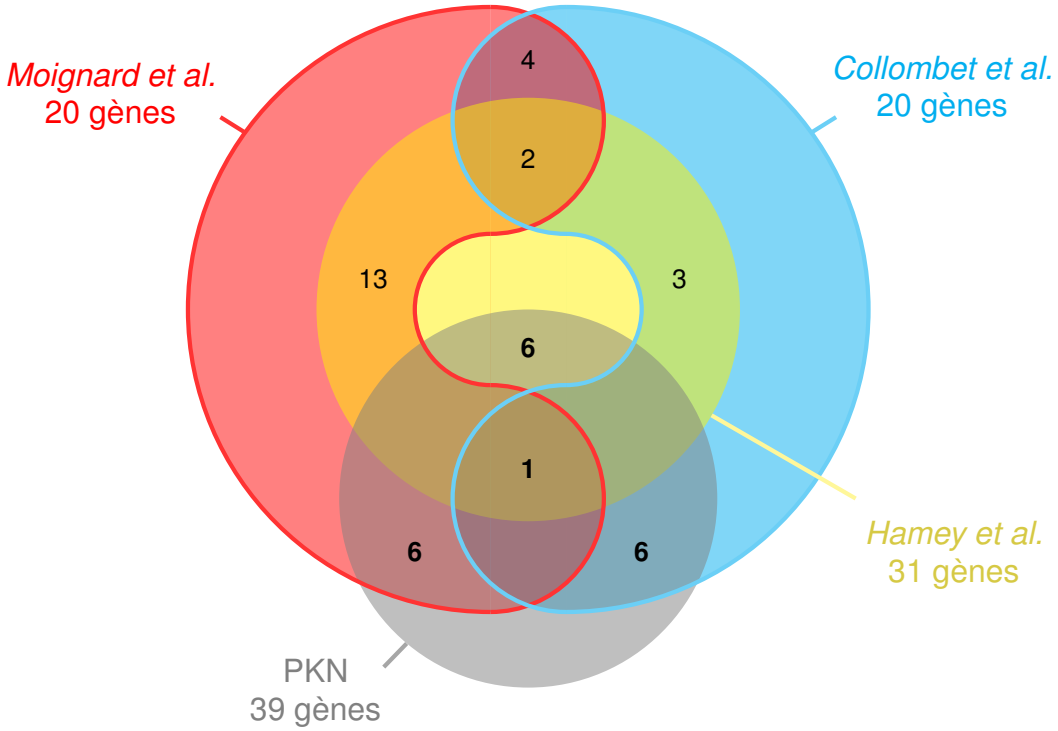


FIGURE 5.20 – Diagramme de Venn présentant le nombre de gènes en commun entre le PKN construit avec BoNesis et les modèles *Hamey et al.* [Hamey et al., 2017], *Moignard et al.* [Moignard et al., 2015] et *Collombet et al.* [Collombet et al., 2017].
Les 3 gènes communs à *Hamey et al.* et *Collombet et al.* : ETS1, IKZF1, RUNX1.
Les 4 gènes communs à *Moignard et al.* et *Collombet et al.* : ETS1, GFI1, IKZF1, SPI1.
Les 13 gènes communs à *Hamey et al.* et *Moignard et al.* : CBFA2T3, ERG, ETS1, FLI1, GATA1, GFI1B, HHEX, HOXB4, IKZF1, LMO2, LYL1, MYB, NFE2.
Le gène commun à tous les modèles et à notre PKN : IKZF1.

seulement des termes synonymes entre les ontologies mais également de la hiérarchie des termes au sein des ontologies, avec des termes allant du processus le plus spécifique à celui le plus général comme l'illustre la figure 5.21.

En effet, l'analyse d'enrichissement fonctionnel peut enrichir des termes qui sont synonymes entre plusieurs ontologies ainsi que des termes fortement apparentés étant donné la hiérarchie au sein des ontologies. Il est alors difficile de repérer, au sein du résultat, des processus à la fois distincts et représentatifs. Metascape se distingue des autres portails qui proposent de l'analyse d'enrichissement car, à l'issue de l'enrichissement, il regroupe automatiquement les termes "similaires" afin de former des groupes mettant davantage en évidence les différents processus biologiques enrichis. Pour cela, il intègre un clustering hiérarchique (sur la base d'un calcul de similarités via un score de Kappa [Cohen, 1960]) afin, d'abord, de regrouper hiérarchiquement les termes dans un arbre, pour ensuite transformer en groupes de termes similaires les sous-arbres ayant un score de similarité supérieur à un seuil défini. Le terme le plus significatif statistiquement au sein de chaque groupe est choisi pour représenter le groupe.

Sur notre ensemble de 39 gènes retenus par BoNesis pour constituer le PKN, le clustering fait très nettement

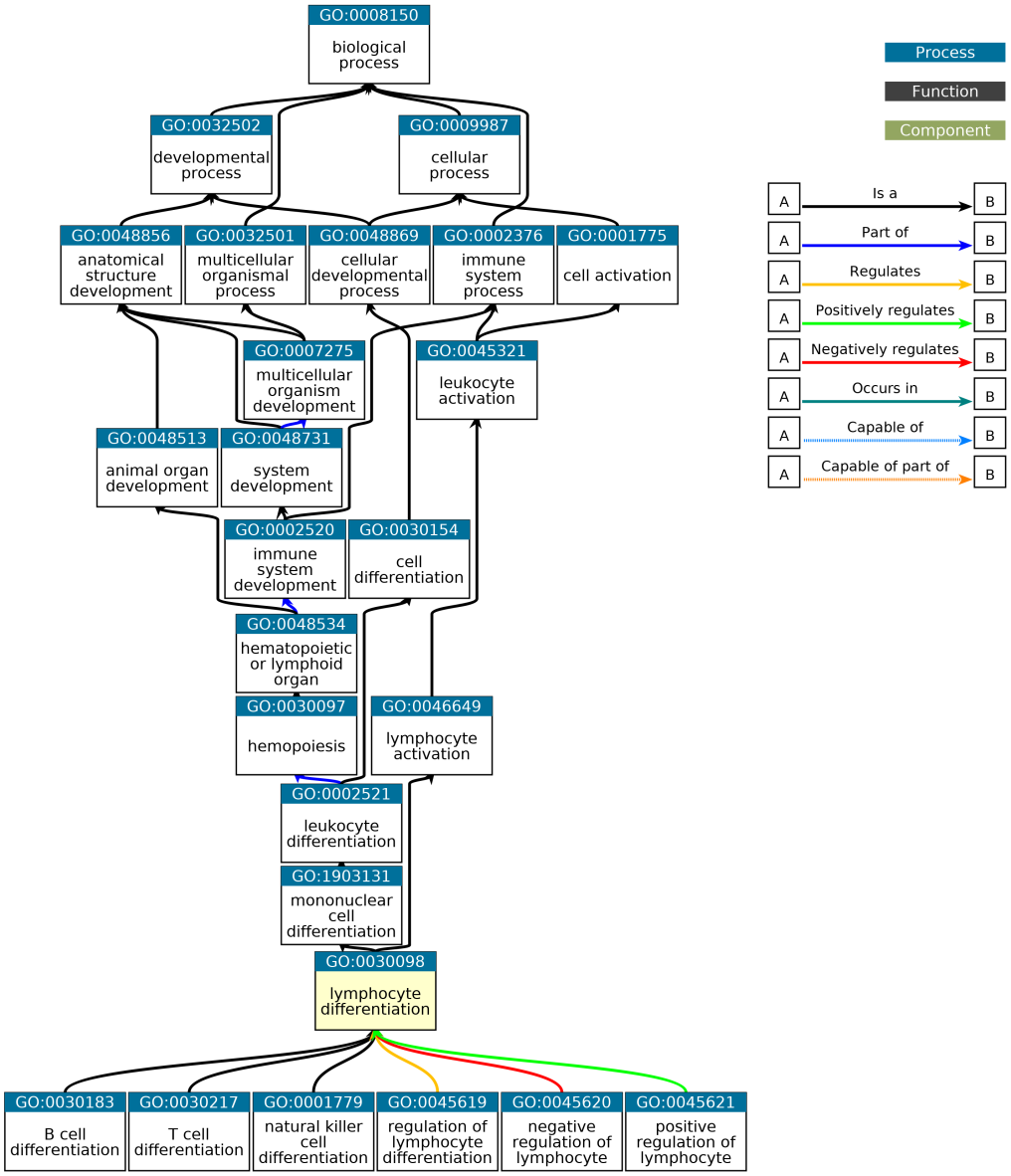


FIGURE 5.21 – DAG montrant, pour le terme *lymphocyte differenciation*, à la fois ses termes "enfants" (en dessous du nœud) et ses "ancêtres" (au-dessus du nœud) au sein de la *Gene Ontology*.

ressortir un enrichissement en termes en lien avec l'hématopoïèse, comme le montre la figure 5.22 avec le top 20 des ensembles de termes similaires statistiquement les plus représentatifs de l'ensemble de gènes. Parmi ces 20, cinq groupes sont en lien avec l'hématopoïèse avec, tout d'abord, le terme "hemopoiesis" en tant que processus biologique associé au plus grand nombre de gènes du PKN (19/39), immédiatement suivi du terme légèrement plus spécialisé "regulation of hemopoiesis" (15/39). On trouve ensuite "mononuclear cell differentiation" (12/39), "regulation of leukocyte differentiation" (10/39) et "B cell activation" (6/39). En remontant à des termes plus généraux que ceux mis en évidence par le clustering (dont le seuil de similarité choisi fait le compromis entre précision et représentativité des différentes fonctions biologiques), on remarque l'association de 30 des 39 gènes avec "cell differentiation" et 24 avec le terme plus spécialisé "regulation of cell differentiation".

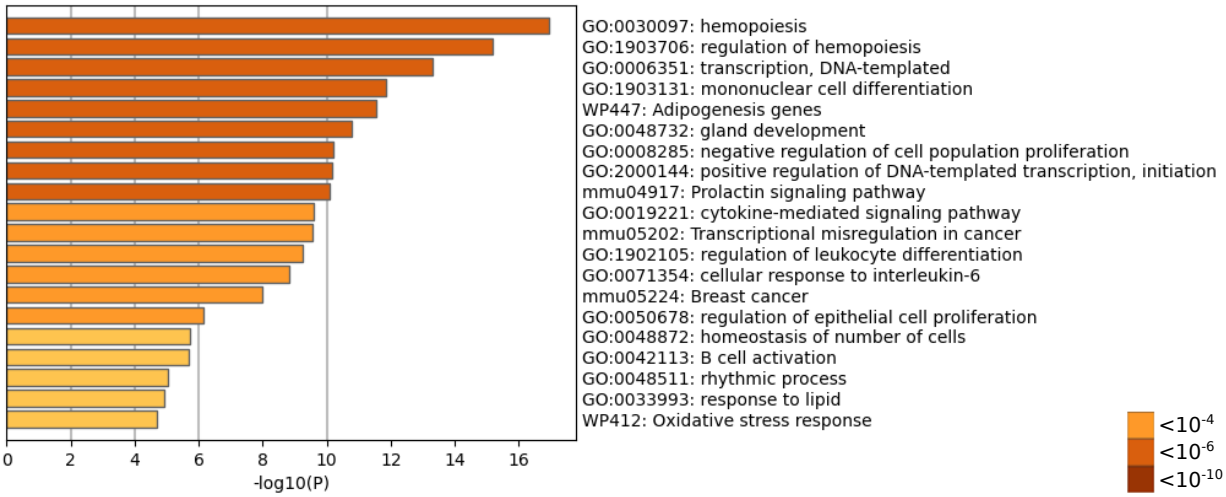


FIGURE 5.22 – Histogramme des 20 groupes de termes les plus enrichis à partir de la liste des 39 gènes constituant le PKN, colorés selon 3 seuils de p-valeurs et avec, par groupe, le terme représentatif du groupe.

5.3.3 Énumération et analyses des modèles

Les données considérées pour la modélisation sont les suivantes :

- les observations construites à partir des données *single-cell*,
- l'ensemble des contraintes existentielles et universelles présentées en 5.3.1.3 afin de décrire le comportement observé,
- le PKN obtenu suite à la sélection des composants (figure 5.19), donnant le sur-ensemble des arêtes possibles.

BoNesis a été utilisé pour énumérer les réseaux booléens dont la dynamique est compatible avec l'ensemble de ces données. Étant donné l'échelle des données considérées, il est fréquent que les observations ne soient pas assez contraignantes au regard du PKN considéré pour éviter un très grand nombre de réseaux booléens compatibles. En effet, le faible nombre de conditions d'observations ne permet souvent pas de discriminer les différentes combinaisons logiques (ET/OU) entre les régulateurs d'un composant ; or ce nombre de combinaisons est exponentiel suivant le nombre de régulateurs. Le nombre de modèles possibles peut donc exploser à cause de quelques composants. Pour autant, d'autres composants peuvent avoir des fonctions identiques dans l'ensemble des modèles énumérés et des motifs d'interactions peuvent être hautement partagés, révélant des éléments clés de la régulation de ce processus. Pour permettre ce type d'analyses, BoNesis peut énumérer les modèles avec une heuristique de *diversité* afin d'obtenir un échantillon de modèles *divers* tel que présenté en 5.1.2.2. Dans le cadre de la modélisation de l'hématopoïèse à partir des données *single-cell*, bien que le domaine des interactions considéré ait été considérablement réduit, nous sommes encore dans cette situation de données de grande taille avec certains composants sans lien avec le comportement à modéliser. Nous avons donc énuméré 1 000 réseaux booléens divers, tous compatibles avec le PKN et le comportement observé décrit par l'ensemble des contraintes existentielles et universelles présentées en 5.3.1.3.

5.3.3.1 Quelles sont les propriétés dynamiques des modèles ?

Étant donné la stratégie de synthèse de modèles de BoNesis, on est assuré que les 1 000 modèles respectent la différenciation telle que décrite par les contraintes listées en 5.3.1.3, avec chacune des trois voies de différenciation aboutissant à un point fixe qui n'est pas atteignable depuis les autres voies.

Nous avons donc la garantie qu'il existe une configuration compatible avec S1 à partir de laquelle on atteint une configuration compatible avec S0 qui peut elle-même atteindre, d'une part, une configuration compatible avec S2 qui est un point fixe et, d'autre part, une configuration compatible avec S3. À partir de cette dernière, on atteint des configurations compatibles avec S4 et S5 qui sont des points fixes. Ces propriétés dynamiques correspondent aux contraintes d'atteignabilités positives et points fixes présentées sur les figures 5.14 et 5.15.

Lister les attracteurs d'un réseau booléen est une analyse très courante lors de l'analyse de modèle, étant donné qu'ils modélisent usuellement des états stables du système biologique étudié. Dans le cadre de cette application, nous sommes également assurés que, depuis la configuration compatible avec S1, les seuls points fixes atteignables sont les configurations compatibles avec S2, S4 et S5 précédemment citées. Tandis que depuis la configuration compatible avec S3, les seuls points fixes atteignables sont les configurations compatibles avec S4 et S5. Ces propriétés dynamiques correspondent aux contraintes d'universalité des points fixes atteignables présentées en figure 5.17. De ce fait, les seules inconnues de cette modélisation à propos des états stables portent sur la présence d'attracteurs non points fixes, c'est-à-dire d'attracteurs cycliques, ainsi que sur le nombre et les caractéristiques des attracteurs au-delà des sous-parties de la dynamique correspondant aux configurations atteignables depuis S1 et S3. Nous n'avions en effet pas d'intérêt particulier à contraindre les points fixes sur la globalité de la dynamique. Concernant les attracteurs au sens plus large, c'est-à-dire en considérant les attracteurs cycliques, il est possible de s'assurer de la constance de marqueurs grâce à la contrainte de confinement mais je n'ai pas développé de contrainte universelle qui garantirait des marqueurs sur l'ensemble des attracteurs d'un réseau booléen quelle que soit leur nature. J'ai donc listé les attracteurs présents dans la dynamique de chacun des 1000 modèles : tous ne contiennent que 3 attracteurs. Ces attracteurs sont les points fixes évoqués ci-dessus, c'est-à-dire les configurations compatibles avec S2, S4 et S5 qui respectent les autres contraintes dans lesquelles elles sont impliquées. J'ai réalisé le calcul des attracteurs de chaque réseau booléen en sémantique *Most Permissive* grâce au package python *mpbn* développé par Loïc Paulevé.

5.3.3.2 Quelles sont les interactions présentes dans les différents modèles ?

Analyser la variabilité des fonctions présentes dans les différents modèles est une étape importante pour se faire une idée de la forme des modèles obtenus et des composants et interactions d'importance pour le comportement modélisé.

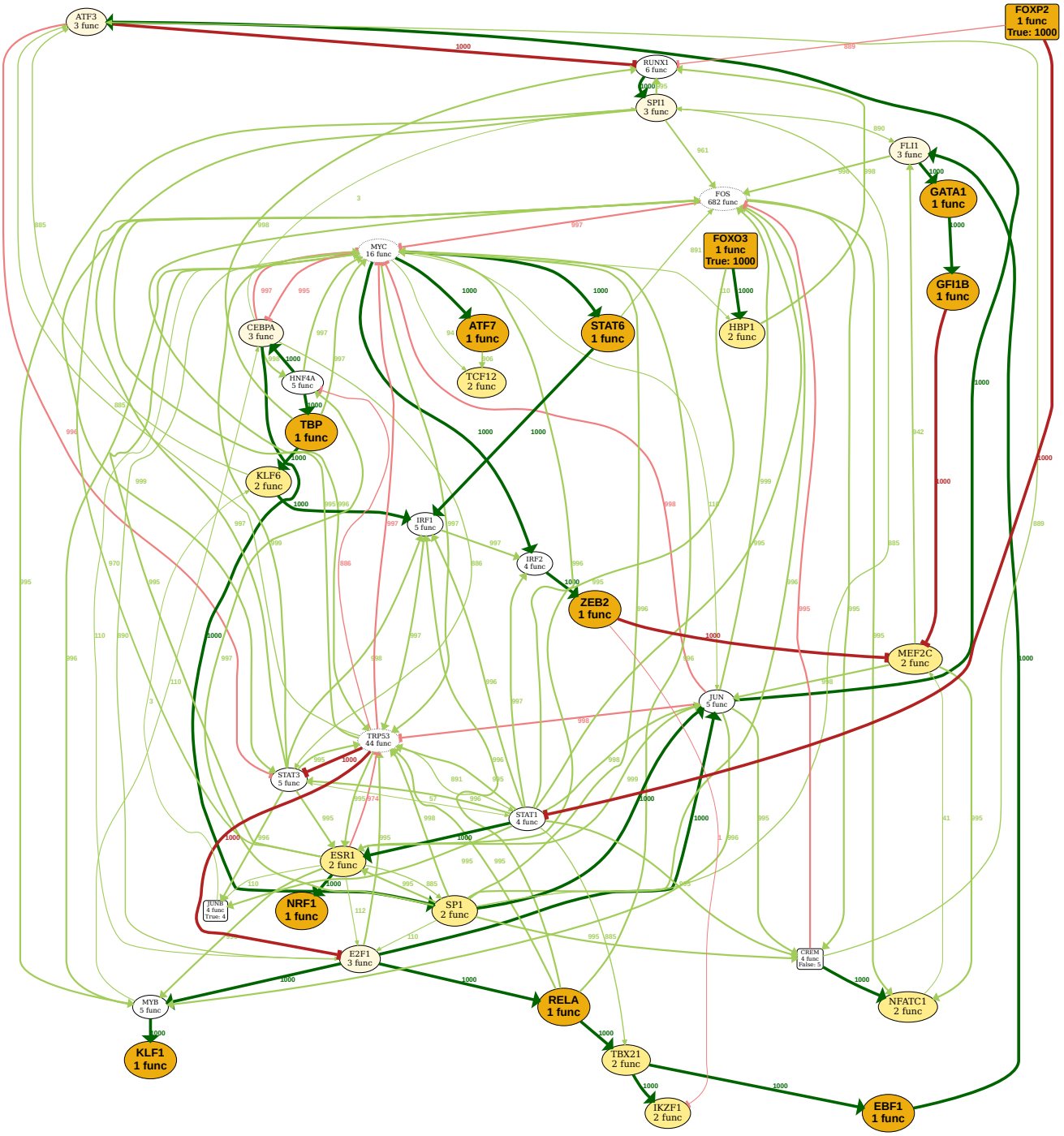


FIGURE 5.23 – Graphe d'interactions synthétisant la structure des 1000 modèles. Les composants sont colorés selon un gradient de jaune qui suit leur variabilité, avec le nombre de fonctions différentes possibles pour chaque composant précisé en étiquette. Les composants ayant une fonction constante dans au moins un modèle sont symbolisés par une icône rectangulaire, avec précision de la valeur de la fonction constante (True : 1, False : 0). Chaque arc est étiqueté par le nombre de modèles (parmi les 1000) qui le possèdent dans leur graphe d'interactions. Les arcs présents dans l'ensemble des 1000 modèles sont en vert et rouge foncés. Un arc activateur est symbolisé par une extrémité en flèche, un arc inhibiteur par une extrémité en "T". Figure agrandissable sur stephaniechevalier.github.io/files/IGstat.pdf

La figure 5.23 réunit les résultats de cette comparaison des fonctions présentes dans les 1000 modèles obtenus. On y remarque que :

- 12 composants ont une fonction identique dans l'ensemble des modèles (ATF7, EBF1, FOXO3, FOXP2, GATA1, GFI1B, KLF1, NRF1, RELA, STAT6, TBP, ZEB2).
- 33 arcs sont communs à l'ensemble des modèles.
- L'essentiel de la variabilité est situé sur le composant FOS, avec 682 fonctions différentes parmi les 1000 modèles. Il est suivi par TRP53 avec 44 fonctions différentes, puis MYC (16 fonctions), RUNX1 (6 fonctions), HNF4A/IRF1/JUN/MYB/STAT3 (4 fonctions).

Le tableau 5.3 présente la répartition du nombre d'arcs suivant le nombre de modèles au sein desquels ils sont présents. On remarque une discontinuité : aucun arc n'est présent dans au moins 113 modèles tout en l'étant dans au plus 884 modèles. Cette répartition laisse penser que plusieurs familles de modèles partageant des propriétés structurelles existent parmi l'ensemble obtenu.

nombre d'arcs	**nombre de modèles (X)**
33	X = 1000
60	X $\geq$ 995 (dans au moins 995 modèles)
111	X $\geq$ 885
15	41 $\leq$ X $\leq$ 112
3	X $\leq$ 3 (dans au plus 3 modèles)

Tableau 5.3 – Nombre d'arcs suivant différentes bornes du nombre de modèles.

5.3.3.3 Quelle est l'hétérogénéité des modèles parmi les 1000 ?

Afin de pouvoir comparer les modèles, j'ai introduit la notion de distance entre réseaux booléens de même dimension. Elle correspond au nombre de composants ayant une fonction différente dans ces deux réseaux booléens.

Définition 5.3.1 (Distance *inter-réseaux booléens*)**.**
Étant donné f *et* g *deux réseaux booléens de dimension* n, $d(f,g) = \sum_{i=1}^{n} \mathbb{1}_{f_i \neq g_i}$.

Par exemple, prenons $f = (f_1(x) = x_1 \vee x_2, f_2(x) = x_3, f_3(x) = \neg x_1)$ et $g = (g_1(x) = x_1 \wedge x_2, g_2(x) = 1, g_3(x) = \neg x_1)$: dans ce cas, $d(f,g) = 2$.

Sur la base de la matrice de distance obtenue, j'ai confronté deux analyses afin de visualiser et identifier d'éventuels groupes de modèles similaires :

- **le positionnement multidimensionnel**, communément appelé MDS *(MultiDimensional Scaling)*, qui est un cas particulier d'analyse multivariée et qui nécessite une fonction de distance. La démonstration que d est bien une fonction de distance est en annexe A.
- **le clustering hiérarchique ascendant**, communément appelé *agglomerative clustering*.

Selon la graine aléatoire utilisée pour le MDS, j'obtiens 3 à 4 groupes de modèles. Deux d'entre eux sont toujours

strictement identiques, contenant 885 et 110 modèles. Les 5 modèles restants apparaissent soit regroupés ensemble, soit répartis en 2 groupes dont la composition varie selon la graine aléatoire utilisée :

- de taille 2 et 3 : $[0, 1]$ et $[93, 675, 966]$
- de taille 1 et 4 : $[1]$ et $[0, 93, 675, 966]$

La méthode du clustering hiérarchique requiert de préciser le nombre de groupes à créer. En fixant cet objectif à 3, les groupes obtenus sont identiques à ceux mis en évidence par le MDS. Lorsqu'on augmente à 4, la division supplémentaire porte à nouveau sur les 5 modèles mis à l'écart des 2 groupes majoritaires, mais selon un nouveau découpage par rapport à ceux observés en MDS : $[675]$ et $[0, 1, 93, 966]$.

Étant donné la constance de la division en 3 groupes quel que soit l'algorithme utilisé, c'est ce découpage que j'ai considéré pour l'analyse de l'ensemble des 1000 réseaux booléens modèles de l'hématopoïèse. Le nuage de points présenté sur la figure 5.24 est le résultat du MDS et le dendrogramme en figure 5.25 est le résultat du clustering hiérarchique ascendant. Comme dit précédemment, les 3 groupes obtenus par l'une et l'autre des méthodes sont identiques. Les couleurs utilisées sur ces deux graphiques ont de ce fait été mises en lien : les réseaux booléens classés dans le groupe orange mis en évidence par le MDS sont les réseaux booléens classés dans le groupe orange du clustering hiérarchique, et il en va de même pour les groupes vert et rouge.

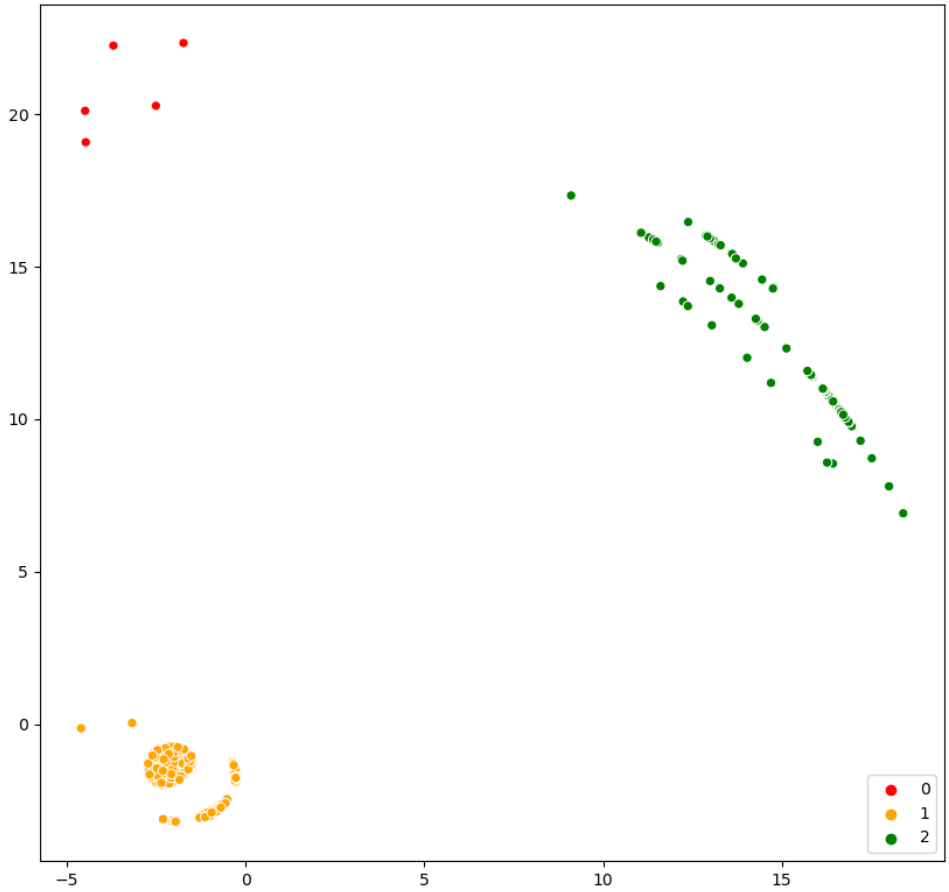


FIGURE 5.24 – Le clustering obtenu par MDS met en évidence 3 groupes.

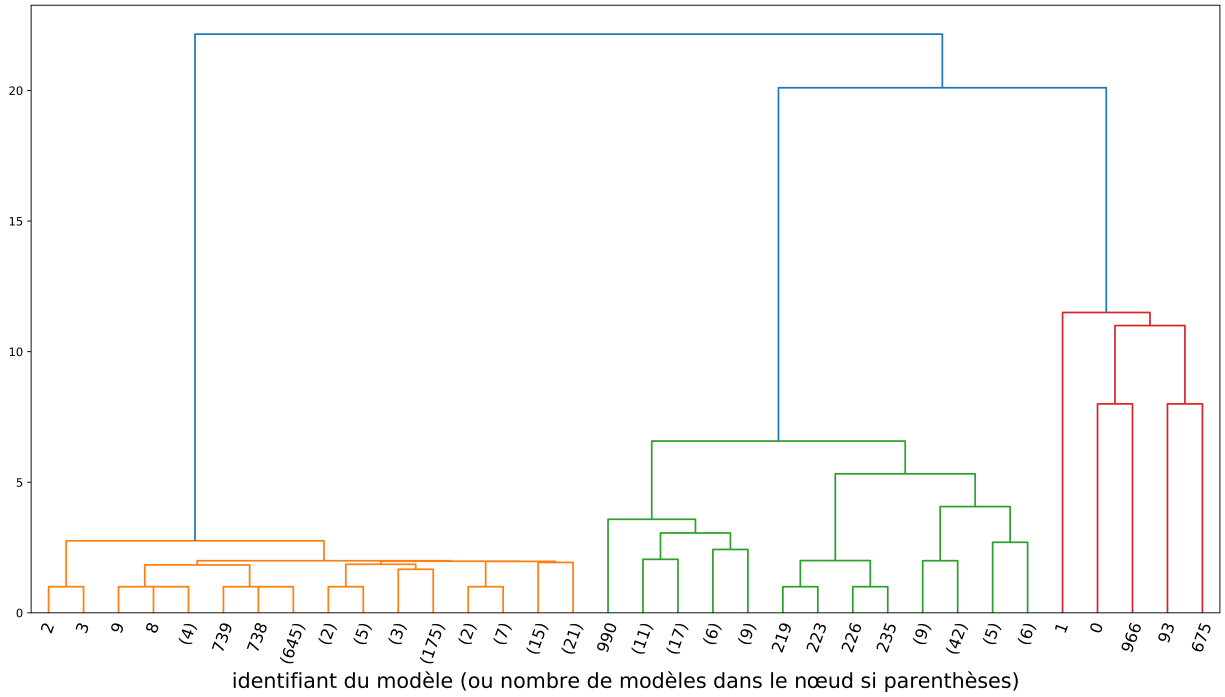


FIGURE 5.25 – Le dendrogramme du clustering obtenu par la méthode hiérarchique ascendante. Les 3 groupes sont exactement ceux observés en MDS.

Particularités des 3 groupes de modèles :

Au sein du groupe orange, la variabilité des 885 réseaux booléens ne concerne que 4 composants au total : CREM, FOS, IKZF1 et TRP53. FOS est de loin le composant le plus variable avec 661 fonctions différentes, suivi par TRP53 avec 28 fonctions, puis CREM et IKZF1 avec 2 fonctions.

Au sein du groupe vert, la variabilité des 110 réseaux booléens concerne 12 composants : FOS (17 fonctions), TRP53 (11), MYC (10), STAT1 (4), FLI1 (3), HNF4A, IRF2, KLF6, MEF2C, MYB, RUNX1, TCF12 (2). En comparant avec les 4 composants variables au sein du groupe orange, on retrouve FOS et TRP53 mais ni CREM ni IKZF1.

Au sein du groupe rouge, bien qu'il ne contienne que 5 réseaux booléens, on observe des fonctions différentes pour 16 de ses composants : MYC et TRP53 (5 fonctions), FOS (4), IRF1, JUN, RUNX1 et STAT3 (3), ATF3, E2F1, HNF4A, IRF2, JUNB, MEF2C, MYB, SPI1, TCF12 (2). Ce groupe de modèles est donc beaucoup plus hétérogène que les deux groupes majoritaires.

Cette comparaison intra et inter-groupes met particulièrement en évidence les informations suivantes :

- La variabilité sur FOS et TRP53 est présente dans l'ensemble des modèles quelle que soit leur groupe.
- Le groupe orange est le plus homogène alors qu'il est de loin le plus grand avec 885 réseaux booléens. On note par ailleurs qu'il est le seul à avoir une variabilité sur les composants CREM et IKRF1.
- Le groupe vert est le seul à avoir une variabilité sur les composants FLI1, KLF6 et STAT1.

- Le groupe rouge est le plus hétérogène alors qu'il est de loin le plus petit avec seulement 5 modèles. Il est le seul à avoir une variabilité sur ATF3, E2F1, IRF1, JUN, JUNB, SPI1 et STAT3.

Lorsqu'on exclut les 5 modèles appartenant au groupe rouge, le nombre de composants ayant une fonction identique dans tous les modèles passe de 12 à 25, le nombre d'arcs concernés passant lui de 33 à 60. Ces composants sont : ATF3, ATF7, CEBPA, E2F1, EBF1, ESR1, FOXO3, FOXP2, GATA1, GFI1B, HBP1, IRF1, JUN, JUNB, KLF1, NFATC1, NRF1, RELA, SP1, SPI1, STAT3, STAT6, TBP, TBX21 et ZEB2.

Les règles de 3 modèles issus de chacun des groupes sont présentées dans le tableau en annexe B.

5.3.3.4 Conclusion et perspectives pour l'étude de l'hématopoïèse

À partir des 1 001 composants extraits de la base de données publiques DoRothEA, BoNesis a permis de construire un PKN en éliminant 962 composants non pertinents au regard des données sur le comportement à modéliser. Sur la base de ce domaine d'interactions de 39 gènes et 137 arcs, 1 000 modèles diversifiés ont été énumérés.

En comparant ces 1 000 réseaux booléens compatibles, on remarque 12 composants dont la fonction est strictement identique dans l'ensemble des modèles quel que soit le groupe auquel ils appartiennent. Parmi eux, GATA1 et GFI1B sont communs aux modèles [Moignard et al., 2015] et [Hamey et al., 2017], et EBF1 est commun au modèle [Collombet et al., 2017]. Liés à ces fonctions, 33 arcs sont communs à l'ensemble des modèles.

L'application de méthodes de clustering, sur la base d'une distance comptant simplement le nombre de fonctions différentes entre les modèles, met en évidence 3 groupes de modèles différents. Le groupe majoritaire rassemble 885 modèles très fortement similaires puisque seuls 4 composants peuvent avoir des fonctions différentes. Parmi ces 4 composants, deux sont en fait très conservés puisque seules 2 fonctions similaires sont possibles. La variabilité se concentre en fait très fortement sur les 2 autres composants, FOS et TRP53, à l'instar de ce qui est observé sur l'ensemble des 1 000 modèles. Cette observation nourrit la question de la pertinence de la présence de ces gènes au sein des modèles de la différenciation hématopoïétique ou de celle du résultat de la binarisation au sein des différentes observations. Les données single-cell ne fournissant pas d'information d'expression pour TRP53, ce gène n'a pas été directement contraint, à l'inverse de FOS dont la figure 5.26 montre le résultat de la binarisation de son expression dans les 6 observations construites le long de la trajectoire de différenciation. La figure 5.27 montre les valeurs d'expression brutes normalisées de FOS, alignées le long des 3 voies de différenciation possibles. Il n'y a pas de lien clairement observable entre son expression et le type cellulaire ; le fait que la méthode de binarisation ait conclu à des valeurs pour l'ensemble des observations créées pourrait être un artefact lié à un nombre insuffisant de cellules incluses pour construire chaque observation.

En analysant les groupes mis en évidence par les méthodes de clustering, on remarque qu'en ne considérant que les groupes orange et vert (995 des 1 000 modèles) du clustering montré en figures 5.24 et 5.25, 25 des 39

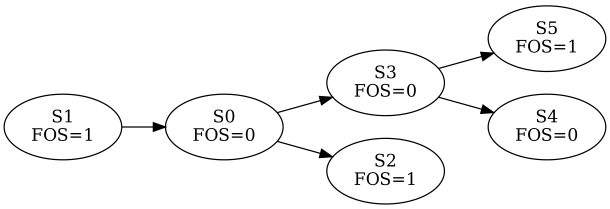


FIGURE 5.26 – Valeurs binarisées de FOS au sein des 6 observations.

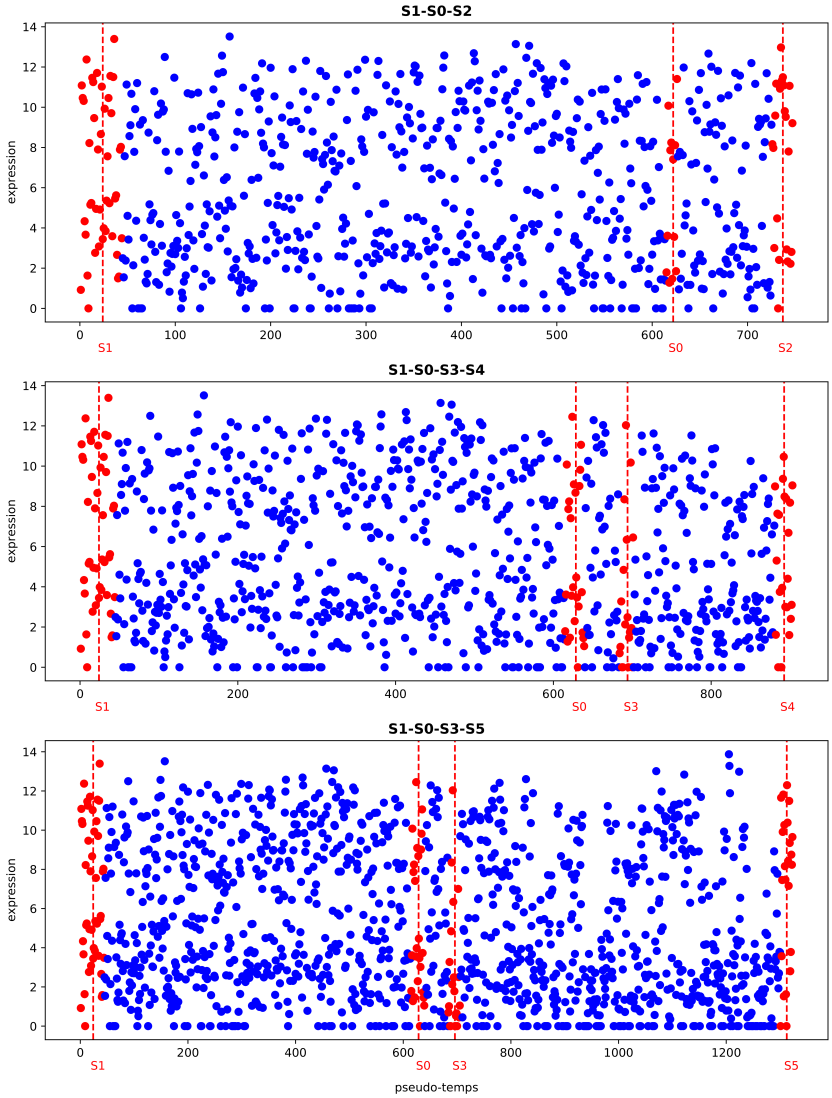


FIGURE 5.27 – Valeurs d'expression normalisées de FOS, alignées le long des 3 voies de différenciation. Les cellules affichées en rouge sont celles retenues pour le calcul de la binarisation afin de construire les différentes observations.

composants ont une fonction identique dans l'ensemble des modèles, et le nombre d'arcs conservés monte à 60. Pour autant, l'intersection avec les modèles de l'état de l'art augmente uniquement au regard de celui de [Collombet et al., 2017], avec l'ajout de CEBPA et SPI1 à EBF1. Étudier ces interactions hautement conservées pourrait permettre d'étendre les connaissances sur ce processus en repérant de nouveaux acteurs et motifs régulateurs de la différenciation.

Résumé du chapitre 5

Je présente dans ce chapitre l'outil d'inférence de réseaux booléens développé avec les contraintes décrites au chapitre 4. Cet outil, nommé BoNesis, aborde la modélisation comme un problème de satisfiabilité booléenne. La stratégie consiste à confronter un graphe d'interactions avec des observations de composants biologiques, couramment des mesures d'expression de gènes, évoluant au cours d'un processus dont les propriétés dynamiques sont décrites sur la base du formalisme présenté au chapitre 3. Cette approche permet deux utilisations complémentaires de BoNesis : l'énumération exhaustive des modèles, mais également la sélection de composants afin d'aider à la construction d'un *Prior Knowledge Network* en lien avec le comportement observé, étape réalisée en amont de l'énumération des modèles lorsque le graphe d'interactions considéré est directement issu d'une base de données d'interactions par exemple. Lors d'une énumération partielle des modèles, une fonction de BoNesis permet d'augmenter la diversité entre les solutions afin d'avoir un sous-ensemble de modèles diversifiés.

En combinant les contraintes présentées au chapitre 4 d'atteignabilité positive et négative, de confinement et de point fixe (potentiellement universel), BoNesis peut exploiter des données de différentes natures (séries temporelles *(bulk data)* / pseudo-temps *(single-cell data)* / états stables (*steady-state data*, gènes marqueurs) / mutations *(perturbation data)* / conditions expérimentales (médicament, ...)) afin de modéliser la richesse de comportements biologiques tels que la différenciation cellulaire. Via des contraintes d'atteignabilité positive entre les observations, il est possible de spécifier l'ordonnancement des observations et ainsi modéliser l'évolution d'un état cellulaire. La contrainte de point fixe répond aux interprétations usuelles des *steady-state expression data* en décrivant une stabilité totale, tandis que la contrainte de confinement sur une observation permet de modéliser la stabilité d'un marqueur de phénotype. Il est également possible de spécifier l'expression ou le silence forcé d'un composant afin de modéliser, par exemple, l'effet de mutations ou de l'administration d'un médicament. Enfin, la contrainte d'atteignabilité négative d'une configuration à une autre est un moyen de décrire les bifurcations, qui peuvent également être décrites via l'application de contraintes universelles sur les points fixes atteignables. Cette dernière contrainte est particulièrement adaptée à la modélisation d'états stables résultant de différentes perturbations telles que des mutations.

J'ai participé à deux applications de modélisation de différenciations cellulaires avec BoNesis. La première considère les états stables observés selon différentes mutations. Les modèles sont construits sur la base du graphe d'interactions d'un modèle publié et l'analyse de l'ensemble de modèles inférés met en évidence le potentiel métastatique variable d'une double mutation, variabilité non observée sur le modèle initial. Cette modélisation a été publiée dans [Chevalier et al., 2020]. La seconde application illustre l'applicabilité de la méthode sur des données *single-cell* pour modéliser une différenciation cellulaire. Je présente la stratégie que j'ai adoptée pour exploiter ces données particulières ainsi qu'un graphe d'interactions de plusieurs milliers de gènes issu d'une base de données publique d'interactions. L'énumération partielle de 1000 modèles diversifiés est réalisée à la suite de la construction du PKN à l'aide de BoNesis. L'analyse met en évidence 3 groupes de modèles partageant un même motif d'interactions pour 30% des composants, ainsi qu'une variabilité focalisée sur quelques gènes au sein des groupes.

Conclusion et perspectives

Les modèles dynamiques de réseaux moléculaires sont des outils importants pour la recherche en biologie et en médecine. Ils permettent d'explorer les mécanismes de régulation de comportements biologiques et ils prédisent la réaction du système biologique s'il est confronté à des perturbations.

Les réseaux booléens sont pertinents pour cette modélisation. Ils offrent un haut niveau d'abstraction proche de la granularité des connaissances actuelles sur les interactions moléculaires, et ils peuvent reproduire des dynamiques complexes incluant un nombre important de composants.

Les travaux de cette thèse ont contribué au développement d'une méthode d'inférence de réseaux booléens à partir des données biologiques, nommée BoNesis. À notre connaissance, BoNesis est la seule approche qui permet de modéliser la richesse d'un comportement aussi complexe que la différenciation cellulaire et qui passe à l'échelle des réseaux de régulation couramment considérés en biologie. Nous avons pu aborder des réseaux de plusieurs milliers de composants grâce au calcul de la dynamique du réseau booléen en sémantique *Most Permissive*, sémantique qui apporte également des garanties par rapport aux modèles quantitatifs.

Première contribution : cadre formel de la modélisation des comportements biologiques (chapitre 3). Pour formaliser le comportement de la différenciation cellulaire, je l'ai considéré comme une combinaison de trois comportements biologiques "élémentaires" qu'on peut observer : (i) une évolution cellulaire, (ii) une divergence d'évolutions, (iii) une stabilité cellulaire. J'ai défini ces comportements et la compatibilité d'un réseau booléen avec chacun d'eux selon une interprétation existentielle et une interprétation universelle. Ainsi, j'ai donné une définition de compatibilité d'un réseau booléen avec une liste d'observations afin de décrire l'ordonnancement d'observations lors du suivi d'évolutions cellulaires. J'ai également défini la compatibilité avec une bifurcation afin de décrire une divergence d'évolutions à partir d'une même cellule mère, ainsi que la compatibilité avec des marqueurs de stabilité partielle ou totale, c'est-à-dire des connaissances sur la stabilité de la cellule au moment de son observation. C'est sur la base de ces compatibilités "élémentaires" que j'ai défini la compatibilité d'un réseau booléen avec une différenciation cellulaire dont les observations s'organisent sous la forme d'un arbre de différenciation. À partir de ce formalisme, il est possible de décrire quelles sont les propriétés attendues d'un réseau booléen dit compatible

avec des comportements biologiques variés étant donné les observations et les connaissances qu'on a sur ces comportements.

Seconde contribution : encodage de contraintes en *Answer-Set Programming* afin de contraindre l'inférence automatique à l'ensemble des modèles compatibles avec les données biologiques (chapitre 4). La méthode d'inférence de modèles repose sur la programmation par ensemble-réponse (ASP). La stratégie utilisée consiste à décrire le problème d'inférence comme un problème de satisfiabilité implémenté sous la forme d'un programme logique en ASP. Le programme contient l'encodage et l'évaluation des fonctions d'un réseau booléen, mais également le graphe des interactions qui peuvent composer les fonctions booléennes ainsi que l'ensemble des données décrivant le comportement biologique avec lequel le réseau booléen doit être compatible. Pour qu'il soit possible d'inférer des modèles de différenciations cellulaires, j'ai implémenté des contraintes assurant la compatibilité d'un réseau booléen avec six interprétations possibles des comportements biologiques que j'ai précédemment décrits comme élémentaires. La contrainte d'atteignabilité positive permet de garantir la compatibilité d'un réseau booléen avec des listes d'observations afin de modéliser l'évolution cellulaire. La contrainte d'atteignabilité négative garantit la compatibilité avec des bifurcations afin de modéliser la divergence d'évolutions. Concernant la stabilité cellulaire, il y a la contrainte de point fixe, qui garantit la compatibilité avec des marqueurs de stabilité totale, et la contrainte de confinement, qui garantit la compatibilité avec des marqueurs de stabilité possiblement partielle. Je propose également une contrainte universelle de point fixe, déclinée en point fixe atteignable, qui permet deux interprétations supplémentaires. Elle garantit l'inférence d'un réseau booléen compatible universellement avec un ensemble de marqueurs de stabilité totale, contraignant l'ensemble de la dynamique du réseau booléen ou la partie atteignable depuis une observation donnée. Les contraintes existentielles ont été publiées dans [Chevalier et al., 2019], contribution présentée à la conférence ICTAI 2019, et la contrainte universelle dans [Chevalier et al., 2020], présentée à la conférence CMSB 2020.

Perspectives de cette seconde contribution : La contrainte universelle sur les points fixes garantit qu'il n'existe aucun point fixe incompatible avec les marqueurs de stabilité totale. Cependant, aucune contrainte ne s'applique sur les attracteurs cycliques du réseau booléen. Si on n'interdit pas leur présence au sein des modèles inférés, les attracteurs cycliques peuvent de ce fait décrire un comportement cellulaire stable qui n'est pas souhaité. Une contrainte universelle sur les attracteurs permettrait d'éviter cette situation. Malgré l'intérêt important de cette contrainte, elle n'a pas été abordée pendant la thèse en raison de sa complexité. En effet, la propriété dynamique concernée a une complexité qui dépasse deux niveaux de quantification au sein de la formule booléenne. Cette complexité ne peut pas être abordée en ASP, ce qui demande de réfléchir à une stratégie de recherche de solutions couplant ASP à un autre système de vérification.

Troisième contribution : application de la méthode d'inférence de modèles sur des données biologiques réelles (chapitre 5). Les contraintes implémentées ont permis de développer un outil d'inférence, BoNesis, capable d'inférer des modèles de différenciation cellulaire. Il calcule la dynamique des réseaux booléens avec la sémantique *Most Permissive*. Ce choix permet de considérer les comportements qui sont observés lorsqu'il y a différents seuils d'activation sans recourir à des modèles multivalués ou quantitatifs. De plus, il rend abordable la modélisation de processus biologiques de plusieurs milliers de composants, ce qui permet à BoNesis d'offrir une fonctionnalité connexe à l'inférence de réseaux booléens : la construction du *Prior Knowledge Network* via la sélection de composants pertinents au regard des observations. Définir le PKN est une étape particulièrement sensible de la préparation des données en amont de l'inférence des modèles, et il est désormais possible de la réaliser en confrontant des observations directement à l'ensemble d'une base de données d'interactions. J'ai participé à une première application de BoNesis en collaboration avec l'institut Curie, sur la base d'un modèle booléen précédemment publié d'une voie de signalisation régulant le destin cellulaire dans la progression du cancer. L'objectif a été de comparer les prédictions sur le modèle publié avec celles obtenues en considérant un ensemble de modèles, en évaluant le changement de destins cellulaires sous l'effet de plusieurs mutations. Cette application a mis en évidence un comportement biologique absent de la simulation du modèle publié mais présent dans la simulation de l'ensemble des modèles. La modélisation et l'analyse des modèles obtenus grâce à BoNesis ont été publiées dans [Chevalier et al., 2020]. J'ai également réalisé une modélisation dynamique de l'hématopoïèse afin d'illustrer l'utilisation de BoNesis à partir de données particulières. J'ai en effet exploité des mesures d'expression obtenues via un séquençage transcriptomique *single-cell*, et le PKN a été construit à l'aide de BoNesis à partir des interactions issues d'une base de données publiques. En initiant l'analyse de l'ensemble des modèles inférés, un motif de régulation se distingue quelle que soit la structure des modèles, mettant en évidence des gènes qui ne sont pas présents dans les modèles de ce même comportement dans l'état de l'art. Il est à noter que l'outil BoNesis a déjà été employé par d'autres chercheurs pour réaliser un modèle dynamique de l'hématopoïèse [Hérault et al., 2022]. Pour cette modélisation, ils ont également exploité des observations *single-cell* mais ils ont basé leur *Prior Knowledge Network* sur un modèle préexistant qu'ils ont complété manuellement.

Perspectives de cette troisième contribution : L'inférence de modèles nécessite à la fois une préparation des données biologiques en amont, mais également une analyse post-inférence capable de traiter un ensemble de modèles. Créer un pipeline qui lierait les outils permettant de préparer les données en amont de l'inférence, et qui rendrait compte de l'ensemble des modèles inférés grâce notamment à des méthodes de visualisation, serait une aide importante pour faciliter l'accès à la modélisation. En amont de l'inférence, il permettrait notamment de guider le modélisateur qui souhaite exploiter des mesures d'expression *single-cell* étant donné la préparation particulière qu'implique ce type de données. De plus, les choix réalisés pendant la préparation des données peuvent fortement impacter la structure des modèles inférés avec BoNesis. En particulier pour l'étape de binarisation qui est

cruciale dans la création des observations, coupler BoNesis avec plusieurs méthodes de binarisation permettrait au modélisateur d'évaluer les conséquences de la méthode choisie sur le résultat. En sortie de BoNesis, donner accès à différentes analyses aiderait le modélisateur à explorer l'ensemble de modèles inféré.

Perspectives générales. BoNesis renvoie l'ensemble des réseaux booléens (ou éventuellement un sous-ensemble diversifié) compatibles avec des connaissances pré-établies sur les régulations deux-à-deux entre composants, et qui reproduisent un comportement dynamique spécifié de manière logique. Ce résultat, constitué d'un ensemble de modèles aussi pertinents les uns que les autres, peut être simulé grâce à une extension de MaBoss présentée dans [Chevalier et al., 2020]. Mais en ce qui concerne la structure des modèles inférés, il est nécessaire de développer des analyses d'ensemble de modèles. Dans l'application présentée en 5.3, j'ai pour cela proposé une visualisation sur la base d'un graphe d'interactions (en figure 5.23) sur lequel sont mis en évidence les composants dont les fonctions sont identiques ou fortement conservées parmi les modèles. J'ai également proposé d'explorer les différentes structures de modèles en appliquant un *Multidimensional Scaling* ou un clustering hiérarchique sur la base d'une fonction de distance "élémentaire", comptant le nombre de fonctions locales différentes entre deux modèles. Pour poursuivre l'exploration, une piste serait d'affiner la fonction de distance afin qu'elle rende compte du niveau de similarité entre les fonctions booléennes locales de chaque composant. Comparer les modèles entre eux est important pour mettre en évidence des motifs d'interactions hautement conservés qui sont très probablement au cœur de la régulation du comportement modélisé, mais également des composants dont la grande variabilité peut souligner l'incohérence d'une hypothèse ou un biais dans la préparation des données en amont de l'inférence. Il est également informatif d'étudier les "familles" formées par les modèles partageant une structure similaire. Il est possible que l'une de ces familles apparaisse plus pertinente que les autres au regard de l'expertise du modélisateur sur le système biologique qu'il étudie.

La stratégie de modélisation a été développée dans le but de reproduire un comportement cellulaire décrit via l'évolution d'expression de gènes. Il me semble que cette stratégie peut être étendue aux réseaux causaux souvent utilisés en recherche médicale sur des pathologies encore mal caractérisées. Ce type de réseau implique des données de nature beaucoup plus hétérogène que les réseaux d'interactions de gènes jusqu'à présent abordés. Un tel réseau met en lien des observations cliniques avec des gènes et des protéines, ainsi qu'avec des traitements et des conditions environnementales. Il est souvent représenté comme un réseau organisé sur plusieurs niveaux interagissant entre eux, chaque niveau correspondant à des éléments de nature similaire (les interactions entre gènes constituent par exemple un niveau). Dans ce contexte, l'objectif serait d'inférer des modèles capables de reproduire des données de suivi de patients. J'ai l'intuition qu'utiliser la méthode d'inférence de réseaux booléens dans ce cadre nécessiterait d'adapter le formalisme présenté au chapitre 3 mais que ces adaptations resteraient mineures et que la démarche de modélisation dynamique serait une aide au raisonnement pour la recherche médicale. Les contributions de la thèse permettent d'envisager l'extension des applications car elles apportent une méthodologie générale pour construire des modèles à partir de connaissances et d'observations d'un système.

Annexe A

Distance inter-réseaux booléens

Étant donné :

- un ensemble M de réseaux booléens, tous de dimension n,
- $\forall f \in M, \forall i \leq n$, f_i est la fonction associée au i^e composant du réseau booléen f.

Soit d la fonction définie de la manière suivante : $\forall f, g \in M$, $d(f,g) = \sum_{i=1}^{n} \mathbb{1}_{f_i \neq g_i}$. Je démontre ci-dessous que d est une fonction de distance.

Positivité :

Par définition de la fonction indicatrice, $\mathbb{1}_{f_i \neq g_i} \in \{0,1\}$. La fonction d correspond donc à une somme d'entiers positifs ou nuls. En conséquence, pour tout $(f,g) \in M^2$, on a $d(f,g) \geq 0$.

Séparation :

Pour tout $(f,g) \in M^2$, on a :

$$d(f,g) = 0 \iff \sum_{i=1}^{n} \mathbb{1}_{f_i \neq g_i} = 0 \iff \forall i \in n, \mathbb{1}_{f_i \neq g_i} = 0 \iff \forall i \leq n, f_i = g_i \iff f = g.$$

Symétrie :

Pour tout $(f,g) \in M^2$, on a $d(f,g) = \sum_{i=1}^{n} \mathbb{1}_{f_i \neq g_i} = \sum_{i=1}^{n} \mathbb{1}_{g_i \neq f_i} = d(g,f)$.

Inégalité triangulaire :

Pour tout $(f,g,h) \in M^3$, on veut $d(f,g) \leq d(f,h) + d(h,g)$, c'est-à-dire $d(f,g) - d(f,h) - d(h,g) \leq 0$.

$d(f,g) - d(f,h) - d(h,g) = \sum_{i=1}^{n} \mathbb{1}_{f_i \neq g_i}$ - $\sum_{i=1}^{n} \mathbb{1}_{f_i \neq h_i}$ - $\sum_{i=1}^{n} \mathbb{1}_{h_i \neq g_i} = \sum_{i=1}^{n} \mathbb{1}_{f_i \neq g_i} - \mathbb{1}_{f_i \neq h_i} - \mathbb{1}_{h_i \neq g_i}$

Soit $1 \leq i \leq n$, on sait que $\mathbb{1}_{f_i \neq g_i} - \mathbb{1}_{f_i \neq h_i} - \mathbb{1}_{h_i \neq g_i} > 0$ si et seulement si les trois conditions suivantes sont réunies :

- $\mathbb{1}_{f_i \neq g_i} = 1$, autrement dit $f_i \neq g_i$.
- $\mathbb{1}_{f_i \neq h_i} = 0$, autrement dit $f_i = h_i$.

- $\mathbb{1}_{h_i \neq g_i} = 0$, autrement dit $h_i = g_i$.

Or, si on a $f_i = h_i$ et $h_i = g_i$, alors $f_i = g_i$. En conséquence, pour tout $1 \leq i \leq n$, on obtient $\mathbb{1}_{f_i \neq g_i} - \mathbb{1}_{f_i \neq h_i} - \mathbb{1}_{h_i \neq g_i} \leq 0$. En définitive, $d(f, g) - d(f, h) - d(h, g) \leq 0$ c'est à dire $d(f, g) \leq d(f, h) + d(h, g)$. La fonction d respecte donc bien l'inégalité triangulaire.

En définitive, d est bien une fonction de distance.

Annexe B

Règles de 3 modèles de groupes différents

	modèle 738 (gr. orange)	modèle 226 (gr. vert)	modèle 93 (gr. rouge)
ATF3	$JUN \vee (CREM \wedge KLF6) \vee (CREM \wedge TRP53) \vee (TRP53 \wedge KLF6)$	JUN	$CREM \vee JUN$
ATF7	MYC	MYC	MYC
CEBPA	$\neg MYC \wedge HNF4A$	$(MYB \wedge HNF4A) \vee (\neg MYC \wedge HNF4A)$	$HNF4A$
CREM	$(FOS \wedge JUN \wedge STAT1) \vee (JUN \wedge SP1 \wedge STAT1)$	$(FOS \wedge JUN \wedge STAT1) \vee (FOS \wedge SP1 \wedge STAT1) \vee (JUN \wedge SP1 \wedge STAT1)$	0
E2F1	$\neg TRP53$	$\neg TRP53 \vee (MYC \wedge ESR1) \vee (SP1 \wedge ESR1) \vee (SP1 \wedge MYC)$	$ESR1 \vee \neg TRP53$
EBF1	$TBX21$	$TBX21$	$TBX21$
ESR1	$(SP1 \wedge FOXO3 \wedge STAT1) \vee (STAT3 \wedge FOXO3 \wedge STAT1) \vee (TRP53 \wedge FOXO3 \wedge STAT1) \vee (SP1 \wedge STAT3 \wedge STAT1) \vee (TRP53 \wedge SP1 \wedge STAT1) \vee (TRP53 \wedge STAT3 \wedge STAT1)$	$(SP1 \wedge FOXO3 \wedge STAT1) \vee (STAT3 \wedge FOXO3 \wedge STAT1) \vee (TRP53 \wedge FOXO3 \wedge STAT1) \vee (SP1 \wedge STAT3 \wedge STAT1) \vee (TRP53 \wedge SP1 \wedge STAT1) \vee (TRP53 \wedge STAT3 \wedge STAT1)$	$STAT1$

FLI1	$(EBF1 \wedge MEF2C) \vee (EBF1 \wedge SPI1)$	$(EBF1 \wedge MEF2C) \vee (EBF1 \wedge SPI1)$	$EBF1$
FOS	$(\neg CREM \wedge FLI1 \wedge STAT3) \vee (JUN \wedge FLI1 \wedge ESR1) \vee (SP1 \wedge FLI1 \wedge ESR1) \vee (\neg CREM \wedge JUN \wedge SP1 \wedge STAT1) \vee (TRP53 \wedge SP1 \wedge STAT6 \wedge ESR1) \vee (TRP53 \wedge \neg CREM \wedge SPI1 \wedge SP1 \wedge ESR1 \wedge STAT3) \vee (TRP53 \wedge \neg CREM \wedge JUN \wedge SP1 \wedge SPI1 \wedge FLI1) \vee (TRP53 \wedge \neg CREM \wedge JUN \wedge SP1 \wedge SPI1 \wedge STAT3)$	$STAT1 \vee (FLI1 \wedge STAT3) \vee (JUN \wedge STAT3) \vee (JUN \wedge TRP53 \wedge ESR1) \vee (SPI1 \wedge FLI1 \wedge TRP53) \vee (JUN \wedge SP1 \wedge SPI1) \vee (\neg CREM \wedge JUN \wedge SP1 \wedge FLI1) \vee (JUN \wedge SP1 \wedge FLI1 \wedge ESR1)$	$ESR1 \vee JUN$
FOXO3	1	1	1
FOXP2	1	1	1
GATA1	$FLI1$	$FLI1$	$FLI1$
GFI1B	$GATA1$	$GATA1$	$GATA1$
HBP1	$FOXO3$	$FOXO3 \vee MYC$	$FOXO3$
HNF4A	$(SP1 \wedge CEBPA) \vee (\neg TRP53 \wedge CEBPA)$	$CEBPA \vee SP1$	$SP1$
IKZF1	$TBX21$	$TBX21$	$TBX21$
IRF1	$(RELA \wedge KLF6) \vee (KLF6 \wedge STAT1) \vee (STAT3 \wedge KLF6) \vee (RELA \wedge STAT1) \vee (STAT3 \wedge STAT1) \vee (STAT6 \wedge STAT1) \vee (STAT3 \wedge STAT6)$	$(KLF6 \wedge STAT1) \vee (STAT3 \wedge KLF6) \vee (STAT6 \wedge KLF6) \vee (RELA \wedge STAT1) \vee (STAT3 \wedge STAT1) \vee (STAT6 \wedge STAT1) \vee (STAT3 \wedge STAT6)$	$KLF6 \vee STAT6$
IRF2	$(MYC \wedge IRF1) \vee (STAT1 \wedge IRF1) \vee (MYC \wedge STAT1)$	$(MYC \wedge IRF1) \vee (MYC \wedge STAT1)$	MYC

JUN	$(E2F1 \wedge ESR1) \vee (E2F1 \wedge MEF2C) \vee (MEF2C \wedge ESR1) \vee (SP1 \wedge ESR1) \vee (ESR1 \wedge STAT1) \vee (SP1 \wedge MEF2C)$	$(E2F1 \wedge ESR1) \vee (E2F1 \wedge MEF2C) \vee (MEF2C \wedge ESR1) \vee (MYC \wedge ESR1) \vee (SP1 \wedge ESR1) \vee (ESR1 \wedge STAT1) \vee (SP1 \wedge MEF2C)$	$E2F1 \vee ESR1 \vee SP1$
JUNB	$STAT1 \vee STAT3$	$ESR1 \vee STAT1 \vee STAT3$	1
KLF1	MYB	MYB	MYB
KLF6	TBP	$JUNB \wedge TBP$	TBP
MEF2C	$\neg GFI1B \vee \neg ZEB2$	$\neg GFI1B \vee \neg ZEB2$	$\neg GFI1B \vee NFATC1 \vee \neg ZEB2$
MYB	$ESR1 \vee (JUN \wedge E2F1) \vee (SPI1 \wedge E2F1)$	$(JUN \wedge E2F1) \vee (SPI1 \wedge E2F1) \vee (JUN \wedge ESR1)$	$E2F1$
MYC	$(\neg FOS \wedge \neg JUN \wedge STAT3) \vee (\neg FOS \wedge \neg JUN \wedge E2F1 \wedge HNF4A) \vee (\neg FOS \wedge HNF4A \wedge E2F1 \wedge STAT1) \vee (\neg FOS \wedge E2F1 \wedge STAT3 \wedge HNF4A) \vee (\neg CEBPA \wedge MYB \wedge SP1 \wedge ESR1 \wedge STAT1) \vee (TBP \wedge MYB \wedge HNF4A \wedge ESR1 \wedge E2F1 \wedge \neg FOS) \vee (\neg CEBPA \wedge MYB \wedge E2F1 \wedge \neg FOS \wedge RELA \wedge STAT3 \wedge \neg TRP53) \vee (\neg JUN \wedge \neg CEBPA \wedge SP1 \wedge STAT1 \wedge HNF4A \wedge E2F1 \wedge RELA \wedge STAT3)$	$(\neg TRP53 \wedge \neg CEBPA \wedge ESR1) \vee (\neg FOS \wedge \neg CEBPA \wedge STAT3) \vee (\neg TRP53 \wedge STAT3 \wedge STAT1) \vee (\neg FOS \wedge \neg JUN \wedge \neg TRP53 \wedge ESR1) \vee (\neg CEBPA \wedge TBP \wedge \neg FOS \wedge RELA \wedge \neg TRP53) \vee (TBP \wedge SP1 \wedge HNF4A \wedge \neg FOS \wedge \neg TRP53) \vee (\neg JUN \wedge MYB \wedge STAT1 \wedge \neg FOS \wedge STAT3) \vee (TBP \wedge HNF4A \wedge ESR1 \wedge RELA \wedge STAT3 \wedge \neg TRP53)$	$(SP1 \wedge E2F1) \vee (\neg JUN \wedge STAT3)$
NFATC1	$(CREM \wedge FOS) \vee (CREM \wedge MEF2C)$	$(CREM \wedge FOS) \vee (CREM \wedge MEF2C)$	$CREM$
NRF1	$ESR1$	$ESR1$	$ESR1$
RELA	$E2F1$	$E2F1$	$E2F1$

RUNX1	$(\neg ATF3 \wedge \neg FOXP2) \vee (HBP1 \wedge \neg ATF3) \vee (SPI1 \wedge TBP) \vee (HBP1 \wedge \neg FOXP2 \wedge TBP)$	$TBP \vee (HBP1 \wedge \neg ATF3) \vee (SPI1 \wedge \neg ATF3)$	$\neg FOXP2 \vee TBP \vee (HBP1 \wedge \neg ATF3)$
SP1	$CEBPA \vee ESR1$	$CEBPA$	$CEBPA$
SPI1	$RUNX1 \vee SP1$	$RUNX1$	$CEBPA \vee RUNX1$
STAT1	$TRP53 \wedge \neg FOXP2$	$TRP53 \wedge STAT3 \wedge \neg FOXP2$	$\neg FOXP2$
STAT3	$SPI1 \vee \neg TRP53 \vee (\neg ATF3 \wedge STAT1) \vee (CEBPA \wedge STAT1)$	$SPI1 \vee \neg TRP53 \vee (\neg ATF3 \wedge STAT1)$	$SPI1 \vee STAT1 \vee \neg TRP53$
STAT6	MYC	MYC	MYC
TBP	$HNF4A$	$HNF4A$	$HNF4A$
TBX21	$RELA \vee STAT1$	$RELA$	$RELA$
TCF12	$ATF7$	MYC	MYC
TRP53	$MYC \vee (FOS \wedge \neg JUN \wedge STAT1) \vee (FOS \wedge E2F1 \wedge \neg ESR1 \wedge STAT1) \vee (\neg JUN \wedge SP1 \wedge E2F1 \wedge IRF1) \vee (\neg JUN \wedge E2F1 \wedge STAT3 \wedge IRF1) \vee (FOS \wedge RELA \wedge STAT3 \wedge IRF1) \vee (FOS \wedge \neg JUN \wedge SP1 \wedge STAT3) \vee (FOS \wedge SP1 \wedge \neg ESR1 \wedge IRF1 \wedge STAT3)$	$SP1 \vee (FOS \wedge IRF1) \vee (FOS \wedge RELA \wedge STAT1) \vee (STAT3 \wedge MYC \wedge IRF1) \vee (RELA \wedge STAT3 \wedge \neg JUN \wedge IRF1) \vee (RELA \wedge \neg JUN \wedge MYC \wedge STAT1) \vee (RELA \wedge \neg JUN \wedge STAT3 \wedge MYC) \vee (\neg JUN \wedge MYC \wedge STAT1 \wedge E2F1 \wedge IRF1)$	$SP1 \vee (\neg JUN \wedge \neg ESR1)$
ZEB2	$IRF2$	$IRF2$	$IRF2$

Tableau B.1 – Les règles de 3 modèles appartenant chacun à l'un des groupes mis en évidence par le clustering.

Bibliographie

S. S. Aghamiri and F. Delaplace. Taboon boolean network synthesis based on tabu search. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2021. doi : 10.1109/TCBB.2021.3063817.

M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology : tool for the unification of biology. *Nature Genetics*, 25(1) :25–29, May 2000. ISSN 1546-1718. doi : 10.1038/75556. URL https://doi.org/10.1038/75556.

J. Béal, A. Montagud, P. Traynard, E. Barillot, and L. Calzone. Personalization of logical models with multi-omics data allows clinical stratification of patients. *Frontiers in Physiology*, 9 :1965, 2019. ISSN 1664-042X. doi : 10.3389/fphys.2018.01965. URL https://www.frontiersin.org/article/10.3389/fphys.2018.01965.

V. Cabeli, L. Verny, N. Sella, G. Uguzzoni, M. Verny, and H. Isambert. Learning clinical networks from medical records based on information estimates in mixed-type data. *PLOS Computational Biology*, 16(5) :1–19, 05 2020. doi : 10.1371/journal.pcbi.1007866. URL https://doi.org/10.1371/journal.pcbi.1007866.

T. Chatain, S. Haar, and L. Paulevé. Most Permissive Semantics of Boolean Networks. *CoRR*, abs/1808.10240, 2018.

H. Chen, L. Albergante, J. Y. Hsu, C. A. Lareau, G. Lo Bosco, J. Guan, S. Zhou, A. N. Gorban, D. E. Bauer, M. J. Aryee, D. M. Langenau, A. Zinovyev, J. D. Buenrostro, G.-C. Yuan, and L. Pinello. Single-cell trajectories reconstruction, exploration and mapping of omics data with stream. *Nature Communications*, 10(1) :1903, Apr 2019. ISSN 2041-1723. doi : 10.1038/s41467-019-09670-4. URL https://doi.org/10.1038/s41467-019-09670-4.

S. Chevalier, C. Froidevaux, L. Paulevé, and A. Zinovyev. Synthesis of Boolean Networks from Biological Dynamical Constraints using Answer-Set Programming. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 34–41, 2019. doi : 10.1109/ICTAI.2019.00014.

S. Chevalier, V. Noël, L. Calzone, A. Zinovyev, and L. Paulevé. Synthesis and Simulation of Ensembles of Boolean

Networks for Cell Fate Decision. In *18th International Conference on Computational Methods in Systems Biology (CMSB)*, Online, Germany, 2020. URL https://hal.archives-ouvertes.fr/hal-02898849.

D. P. A. Cohen, L. Martignetti, S. Robine, E. Barillot, A. Zinovyev, and L. Calzone. Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLOS Computational Biology*, 11(11) :1–29, 11 2015. doi : 10.1371/journal.pcbi.1004571. URL https://doi.org/10.1371/journal.pcbi.1004571.

J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1) :37–46, 1960. doi : 10.1177/001316446002000104. URL https://doi.org/10.1177/001316446002000104.

S. Collombet, C. van Oevelen, J. L. Sardina Ortega, W. Abou-Jaoudé, B. Di Stefano, M. Thomas-Chollier, T. Graf, and D. Thieffry. Logical modeling of lymphoid and myeloid cell specification and transdifferentiation. *Proceedings of the National Academy of Sciences*, 114(23) :5792–5799, 2017. ISSN 0027-8424. doi : 10.1073/pnas.1610622114. URL https://www.pnas.org/content/114/23/5792.

I. H. G. S. Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011) :931–945, Oct. 2004. doi : 10.1038/nature03001. URL https://doi.org/10.1038/nature03001.

R. Dahl, S. R. Iyer, K. S. Owens, D. D. Cuylear, and M. C. Simon. The transcriptional repressor GFI-1 antagonizes PU.1 activity through protein-protein interaction. *Journal of Biological Chemistry*, 282(9) :6473–6483, Mar. 2007. doi : 10.1074/jbc.m607613200.

M. Dahlhaus, A. Burkovski, F. Hertwig, C. Mussel, R. Volland, M. Fischer, K.-M. Debatin, H. A. Kestler, and C. Beltinger. Boolean modeling identifies greatwall/mastl as an important regulator in the aurka network of neuroblastoma. *Cancer Letters*, 371(1) :79–89, 2016. ISSN 0304-3835. doi : https://doi.org/10.1016/j.canlet.2015.11.025. URL https://www.sciencedirect.com/science/article/pii/S0304383515007016.

T. Eiter and G. Gottlob. On the computational cost of disjunctive logic programming : Propositional case. *Annals of Mathematics and Artificial Intelligence*, 15(3) :289–323, Sept. 1995. ISSN 1573-7470. doi : 10.1007/BF01536399.

T. Eiter, G. Ianni, and T. Krennwallner. *Answer Set Programming : A Primer*, pages 40–110. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-03754-2. doi : 10.1007/978-3-642-03754-2_2.

F. Font-Clos, S. Zapperi, and C. A. La Porta. Classification of triple-negative breast cancers through a boolean network model of the epithelial-mesenchymal transition. *Cell Systems*, 12(5) :457–462.e4, 2021a. ISSN 2405-4712. doi : https://doi.org/10.1016/j.cels.2021.04.007. URL https://www.sciencedirect.com/science/article/pii/S2405471221001344.

F. Font-Clos, S. Zapperi, and C. A. L. Porta. Classification of triple-negative breast cancers through a boolean network model of the epithelial-mesenchymal transition. *Cell Systems*, 12(5) :457–462.e4, May 2021b. doi : 10.1016/j.cels.2021.04.007. URL https://doi.org/10.1016/j.cels.2021.04.007.

S. Gao, C. Sun, C. Xiang, K. Qin, and T. H. Lee. Learning asynchronous boolean networks from single-cell data using multiobjective cooperative genetic programming. *IEEE Trans. Cybern.*, 52(5) :2916–2930, May 2022.

L. Garcia-Alonso, C. H. Holland, M. M. Ibrahim, D. Turei, and J. Saez-Rodriguez. Benchmark and integration of resources for the estimation of human transcription factor activities. *Genome Research*, 2019. doi : $10.1101/\mathrm{gr}.240663.118$.

M. Gebser, B. Kaufmann, and T. Schaub. Conflict-driven answer set solving : From theory to practice. *Artif. Intell.*, 187 :52–89, 2012.

M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub. Clingo = ASP + control : Preliminary report. *CoRR*, abs/1405.3694, 2014.

M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and S. Thiele. A user's guide to gringo, clasp, clingo, and iclingo, 2019.

M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R. Kowalski, Bowen, and Kenneth, editors, *Proceedings of International Logic Programming Conference and Symposium*, pages 1070–1080. MIT Press, 1988. URL http://www.cs.utexas.edu/users/ai-lab?gel88.

J. Goldfeder and H. Kugler. Bre :in - a backend for reasoning about interaction networks with temporal logic. In L. Bortolussi and G. Sanguinetti, editors, *Computational Methods in Systems Biology*, pages 289–295, Cham, 2019. Springer International Publishing. ISBN 978-3-030-31304-3.

M. Grunstein and D. S. Hogness. Colony hybridization : a method for the isolation of cloned DNAs that contain a specific gene. *Proceedings of the National Academy of Sciences*, 72(10) :3961–3965, Oct. 1975. doi : $10.1073/\mathrm{pnas}.72.10.3961$. URL https://doi.org/10.1073/pnas.72.10.3961.

Y. Hamdi, M. Boujemaa, M. Ben Rekaya, C. Ben Hamda, N. Mighri, H. El Benna, N. Mejri, S. Labidi, N. Daoud, C. Naouali, O. Messaoud, M. Chargui, K. Ghedira, M. S. Boubaker, R. Mrad, H. Boussen, S. Abdelhak, and the PEC Consortium. Family specific genetic predisposition to breast cancer : results from tunisian whole exome sequenced breast cancer cases. *Journal of Translational Medicine*, 16(1) :158, Jun 2018. ISSN 1479-5876. doi : $10.1186/\mathrm{s}12967\text{-}018\text{-}1504\text{-}9$. URL https://doi.org/10.1186/s12967-018-1504-9.

F. K. Hamey, S. Nestorowa, S. J. Kinston, D. G. Kent, N. K. Wilson, and B. Göttgens. Reconstructing blood stem cell regulatory network models from single-cell molecular profiles. *Proceedings of the National Academy of Sciences*, 114(23) :5822–5829, 2017. ISSN 0027-8424. doi : $10.1073/\mathrm{pnas}.1610609114$. URL https://www.pnas.org/content/114/23/5822.

L. Hérault, M. Poplineau, E. Duprez, and É. Remy. A novel boolean network inference strategy to model early

hematopoiesis aging. *bioRxiv*, 2022. doi : 10.1101/2022.02.08.479548. URL https://www.biorxiv.org/content/early/2022/05/08/2022.02.08.479548.

J. Hoggatt and L. Pelus. Hematopoiesis. In *Brenner's Encyclopedia of Genetics*, pages 418–421. Elsevier, 2013. doi : 10.1016/b978-0-12-374984-0.00686-0. URL https://doi.org/10.1016/b978-0-12-374984-0.00686-0.

C. H. Holland, B. Szalai, and J. Saez-Rodriguez. Transfer of regulatory knowledge from human to mouse for functional genomics analysis. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms*, 1863(6) :194431, 2020. ISSN 1874-9399. doi : https://doi.org/10.1016/j.bbagrm.2019.194431. URL https://www.sciencedirect.com/science/article/pii/S1874939919302287. Transcriptional Profiles and Regulatory Gene Networks.

B. Jassal, L. Matthews, G. Viteri, C. Gong, P. Lorente, A. Fabregat, K. Sidiropoulos, J. Cook, M. Gillespie, R. Haw, F. Loney, B. May, M. Milacic, K. Rothfels, C. Sevilla, V. Shamovsky, S. Shorser, T. Varusai, J. Weiser, G. Wu, L. Stein, H. Hermjakob, and P. D'Eustachio. The reactome pathway knowledgebase. *Nucleic Acids Research*, Nov. 2019. doi : 10.1093/nar/gkz1031. URL https://doi.org/10.1093/nar/gkz1031.

C. Jopling, S. Boue, and J. C. I. Belmonte. Dedifferentiation, transdifferentiation and reprogramming : three routes to regeneration. *Nature Reviews Molecular Cell Biology*, 12(2) :79–89, Jan. 2011. doi : 10.1038/nrm3043. URL https://doi.org/10.1038/nrm3043.

M. Kanehisa. KEGG : Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1) :27–30, Jan. 2000. doi : 10.1093/nar/28.1.27. URL https://doi.org/10.1093/nar/28.1.27.

A. C. Kaushik and S. Sahi. Boolean network model for GPR142 against type 2 diabetes and relative dynamic change ratio analysis using systems and biological circuits approach. *Systems and Synthetic Biology*, 9(1-2) :45–54, Mar. 2015. doi : 10.1007/s11693-015-9163-0. URL https://doi.org/10.1007/s11693-015-9163-0.

D. Kleitman. On Dedekind's problem : The number of monotone Boolean functions. *Proceedings of the American Mathematical Society*, 21(3) :677, 1969. doi : 10.2307/2036446.

C. P. Koh, C. Q. Wang, C. E. L. Ng, Y. Ito, M. Araki, V. Tergaonkar, G. Huang, and M. Osato. RUNX1 meets MLL : epigenetic regulation of hematopoiesis by two leukemia genes. *Leukemia*, 27(9) :1793–1802, July 2013. doi : 10.1038/leu.2013.200.

L. Licata, P. Lo Surdo, M. Iannuccelli, A. Palma, E. Micarelli, L. Perfetto, D. Peluso, A. Calderone, L. Castagnoli, and G. Cesareni. SIGNOR 2.0, the SIGnaling Network Open Resource 2.0 : 2019 update. *Nucleic Acids Research*, 48 (D1) :D504–D510, 10 2019. ISSN 0305-1048. doi : 10.1093/nar/gkz949. URL https://doi.org/10.1093/nar/gkz949.

X. Liu, Y. Wang, N. Shi, Z. Ji, and S. He. Gapore : Boolean network inference using a genetic algorithm with novel polynomial representation and encoding scheme. *Knowledge-Based Systems*, 228 :107277, 2021. ISSN

0950-7051. doi : https://doi.org/10.1016/j.knosys.2021.107277. URL https://www.sciencedirect.com/science/article/pii/S0950705121005396.

J. Lobo, J. Minker, and A. Rajasekar. *Foundations of disjunctive logic programming*. MIT press, 1992.

C. Manzoni, D. A. Kia, J. Vandrovcova, J. Hardy, N. W. Wood, P. A. Lewis, and R. Ferrari. Genome, transcriptome and proteome : the rise of omics data and their integration in biomedical sciences. *Briefings in Bioinformatics*, 19(2) : 286–302, 11 2016. ISSN 1477-4054. doi : 10.1093/bib/bbw114. URL https://doi.org/10.1093/bib/bbw114.

E. R. Mardis. A decade's perspective on dna sequencing technology. *Nature*, 470(7333) :198–203, Feb 2011. ISSN 1476-4687. doi : 10.1038/nature09796. URL https://doi.org/10.1038/nature09796.

A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera, and A. Califano. ARACNE : An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(S1), Mar. 2006. doi : 10.1186/1471-2105-7-s1-s7. URL https://doi.org/10.1186/1471-2105-7-s1-s7.

V. Moignard, S. Woodhouse, L. Haghverdi, A. J. Lilly, Y. Tanaka, A. C. Wilkinson, F. Buettner, I. C. Macaulay, W. Jawaid, E. Diamanti, S.-I. Nishikawa, N. Piterman, V. Kouskoff, F. J. Theis, J. Fisher, and B. Göttgens. Decoding the regulatory network of early blood development from single-cell gene expression measurements. *Nature biotechnology*, 33(3) :269–276, Mar 2015. ISSN 1546-1696. doi : 10.1038/nbt.3154. URL https://pubmed.ncbi.nlm.nih.gov/25664528. PMC4374163[pmcid].

A. Montagud, J. Béal, L. Tobalina, P. Traynard, V. Subramanian, B. Szalai, R. Alföldi, L. Puskás, A. Valencia, E. Barillot, J. Saez-Rodriguez, and L. Calzone. Patient-specific boolean models of signalling networks guide personalised treatments. *eLife*, 11, Feb. 2022. doi : 10.7554/elife.72626. URL https://doi.org/10.7554/elife.72626.

A. Navarro and A. Martínez-Murcia. Phylogenetic analyses of the genus aeromonas based on housekeeping gene sequencing and its influence on systematics. *Journal of applied microbiology*, 125(3) :622—631, September 2018. ISSN 1364-5072. doi : 10.1111/jam.13887. URL https://doi.org/10.1111/jam.13887.

S. Nestorowa, F. K. Hamey, B. Pijuan Sala, E. Diamanti, M. Shepherd, E. Laurenti, N. K. Wilson, D. G. Kent, and B. Göttgens. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood*, 128(8) :e20–e31, 08 2016. ISSN 0006-4971. doi : 10.1182/blood-2016-05-716480. URL https://doi.org/10.1182/blood-2016-05-716480.

M. Ostrowski, L. Paulevé, T. Schaub, A. Siegel, and C. Guziolowski. Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming. *Biosystems*, 149 :139 – 153, 2016a. ISSN 0303-2647. doi : 10.1016/j.biosystems.2016.07.009.

M. Ostrowski, L. Paulevé, T. Schaub, A. Siegel, and C. Guziolowski. Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming. *Biosystems.*, 149 :139–153, Nov. 2016b.

R. Palli, M. G. Palshikar, and J. Thakar. Executable pathway analysis using ensemble discrete-state modeling for large-scale data. *PLoS Comput Biol*, 15(9) :e1007317, Sept. 2019.

A. Palma, M. Iannuccelli, I. Rozzo, L. Licata, L. Perfetto, G. Massacci, L. Castagnoli, G. Cesareni, and F. Sacco. Integrating patient-specific information into logic models of complex diseases : Application to acute myeloid leukemia. *Journal of Personalized Medicine*, 11(2) :117, Feb 2021. ISSN 2075-4426. doi : 10.3390/jpm11020117. URL http://dx.doi.org/10.3390/jpm11020117.

L. Paulevé, J. Kolčák, T. Chatain, and S. Haar. Reconciling qualitative, abstract, and scalable modeling of biological networks. *Nature Communications*, 11(1) :4256, Aug 2020. ISSN 2041-1723. doi : 10.1038/s41467-020-18112-5. URL https://doi.org/10.1038/s41467-020-18112-5.

X. Qiu, Q. Mao, Y. Tang, L. Wang, R. Chawla, H. A. Pliner, and C. Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nature Methods*, 14(10) :979–982, Aug. 2017. doi : 10.1038/nmeth.4402. URL https://doi.org/10.1038/nmeth.4402.

M. Razzaq, R. Kaminski, J. Romero, T. Schaub, J. Bourdon, and C. Guziolowski. Computing diverse boolean networks from phosphoproteomic time series data. In M. Češka and D. Šafránek, editors, *Computational Methods in Systems Biology*, pages 59–74, Cham, 2018. Springer International Publishing. ISBN 978-3-319-99429-1.

G. Robertson, M. Hirst, M. Bainbridge, M. Bilenky, Y. Zhao, T. Zeng, G. Euskirchen, B. Bernier, R. Varhol, A. Delaney, N. Thiessen, O. L. Griffith, A. He, M. Marra, M. Snyder, and S. Jones. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods*, 4(8) : 651–657, June 2007. doi : 10.1038/nmeth1068. URL https://doi.org/10.1038/nmeth1068.

M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235) :467–470, 1995. doi : 10.1126/science.270.5235.467. URL https://www.science.org/doi/abs/10.1126/science.270.5235.467.

J. D. Schwab, N. Ikonomi, S. D. Werle, F. M. Weidner, H. Geiger, and H. A. Kestler. Reconstructing boolean network ensembles from single-cell data for unraveling dynamics in the aging of human hematopoietic stem cells. *Computational and Structural Biotechnology Journal*, 19 :5321–5332, 2021. doi : 10.1016/j.csbj.2021.09.012. URL https://doi.org/10.1016/j.csbj.2021.09.012.

S. C. Sealfon and T. T. Chu. *RNA and DNA Microarrays*, pages 3–34. Humana Press, Totowa, NJ, 2011. ISBN 978-1-59745-551-0. doi : 10.1007/978-1-59745-551-0_1. URL https://doi.org/10.1007/978-1-59745-551-0_1.

E. Shapiro, T. Biezuner, and S. Linnarsson. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nature Reviews Genetics*, 14(9) :618–630, Sep 2013. ISSN 1471-0064. doi : 10.1038/nrg3542. URL https://doi.org/10.1038/nrg3542.

G. Stoll, B. Caron, E. Viara, A. Dugourd, A. Zinovyev, A. Naldi, G. Kroemer, E. Barillot, and L. Calzone. MaBoSS 2.0 : an environment for stochastic Boolean modeling. *Bioinformatics*, 33(14) :2226–2228, 03 2017. ISSN 1367-4803. doi : 10.1093/bioinformatics/btx123. URL https://doi.org/10.1093/bioinformatics/btx123.

C. Terfve, T. Cokelaer, D. Henriques, A. MacNamara, E. Goncalves, M. K. Morris, M. v. Iersel, D. A. Lauffenburger, and J. Saez-Rodriguez. CellNOptR : a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Systems Biology*, 6(1) :133, 2012. ISSN 1752-0509. doi : 10.1186/1752-0509-6-133.

R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3) :563–585, 1973. ISSN 0022-5193. doi : https://doi.org/10.1016/0022-5193(73)90247-6. URL https://www.sciencedirect.com/science/article/pii/0022519373902476.

H.-C. Trinh and Y.-K. Kwon. A novel constrained genetic algorithm-based Boolean network inference method from steady-state gene expression data. *Bioinformatics*, 37 :i383–i391, 07 2021. ISSN 1367-4803. doi : 10.1093/bioinformatics/btab295. URL https://doi.org/10.1093/bioinformatics/btab295.

A. Vaginay, T. Boukhobza, and M. Smaïl-Tabbone. Automatic synthesis of boolean networks from biological knowledge and data. In *International Conference of Optimization and Learning, OLA '2021*, Catane, Italy, June 2021. URL https://hal.archives-ouvertes.fr/hal-03256693.

E. L. van Dijk, H. Auger, Y. Jaszczyszyn, and C. Thermes. Ten years of next-generation sequencing technology. *Trends in Genetics*, 30(9) :418–426, 2014. ISSN 0168-9525. doi : https://doi.org/10.1016/j.tig.2014.07.001. URL https://www.sciencedirect.com/science/article/pii/S0168952514001127.

D. Wiedemann. A computation of the eighth dedekind number. *Order*, 8(1) :5–6, 1991. doi : 10.1007/bf00385808.

C. Yeamans, D. Wang, I. Paz-Priel, B. E. Torbett, D. G. Tenen, and A. D. Friedman. C/EBPα binds and activates the PU.1 distal enhancer to induce monocyte lineage commitment. *Blood*, 110(9) :3136–3142, Nov. 2007. doi : 10.1182/blood-2007-03-080291.

Y. Zhou, B. Zhou, L. Pache, M. Chang, A. H. Khodabakhshi, O. Tanaseichuk, C. Benner, and S. K. Chanda. Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. *Nature Communications*, 10 (1) :1523, Apr 2019. ISSN 2041-1723. doi : 10.1038/s41467-019-09234-6. URL https://doi.org/10.1038/s41467-019-09234-6.

P. Zoppoli, S. Morganella, and M. Ceccarelli. TimeDelay-ARACNE : Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics*, 11(1), Mar. 2010. doi : 10.1186/1471-2105-11-154. URL https://doi.org/10.1186/1471-2105-11-154.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Inférence logique de réseaux booléens à partir de connaissances et d'observations de processus de différenciation cellulaire

Mots clés : biologie des systèmes, inférence de modèles dynamiques, réseau booléen, différenciation cellulaire, réseau de régulation génique, answer set programming

Résumé : Les modèles dynamiques sont des outils importants pour l'exploration des mécanismes de régulation en biologie. Les travaux de cette thèse sont guidés par le besoin exprimé en biologie du développement et en cancérologie d'inférer automatiquement des réseaux booléens reproduisant des processus de différenciation cellulaire. En considérant les observations et les connaissances que les modélisateurs ont à disposition, ce mémoire de thèse présente une approche qui permet de modéliser la richesse de ce comportement cellulaire en inférant l'ensemble des réseaux booléens compatibles tout en passant à l'échelle des réseaux de régulation couramment considérés en biologie. Afin de développer cette méthode, les travaux présentés se décomposent en trois contributions principales. La première contribution est la proposition d'un cadre formel sur les propriétés des données collectées pour étudier la différenciation cellulaire. Ce cadre permet de raisonner sur les propriétés dynamiques souhaitées au sein des réseaux booléens pour qu'ils soient compatibles avec ce comportement cellulaire. La deuxième contribution porte sur l'encodage du problème d'inférence de modèles comme un problème de satisfiabilité booléenne dont les solutions sont les réseaux booléens compatibles avec les données biologiques. Pour cela, des contraintes sur la dynamique des réseaux booléens correspondant aux propriétés précédemment formalisées ont été implémentées en programmation logique. La dernière contribution est l'application à des problématiques biologiques réelles de la méthode d'inférence de modèles, nommée BoNesis, qui a été développée grâce aux contraintes créées. Ces applications ont montré l'apport de l'inférence d'ensemble de modèles pour l'analyse de processus et illustré la méthodologie de modélisation, de la préparation des données biologiques à l'analyse des modèles inférés.

Title : Logical inference of Boolean networks from knowledge and observations of cellular differentiation processes

Keywords : systems biology, dynamical model inference, Boolean network, cell differentiation, gene regulatory network, answer set programming

Abstract : Dynamic models are essential tools for exploring regulatory mechanisms in biology. This thesis was guided by the need expressed in oncology and developmental biology to automatically infer Boolean networks reproducing cellular differentiation processes. By considering observations and knowledge that the modelers have at their disposal, this thesis presents an approach that allows to model the richness of this cellular behavior by inferring all the compatible Boolean networks, at the scale of the regulatory networks commonly considered in biology. To develop this method, three main contributions are presented. The first contribution is a formal framework of the properties of data collected to study cellular differentiation. This framework allows reasoning about the desired dynamic properties within Boolean networks to be consistent with this cellular behavior. The second contribution concerns the encoding of the model inference problem as a Boolean satisfiability problem whose solutions are the Boolean networks compatible with the biological data. For this, constraints on the dynamics of Boolean networks corresponding to the previously formalized properties have been implemented in logic programming. The last contribution was to apply to real biological problems the model inference method, named BoNesis, which was developed thanks to the constraints. These applications showed the benefit of inferring a set of models for the process analysis and illustrated the modeling methodology, from the preparation of biological data to the analysis of the inferred models.