Hybrid, logical and linear modelling to predict in silico the effect of perturbations on metabolism

Sophie Le Bars

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Nantes Université

THÈSE DE DOCTORAT DE

NANTES UNIVERSITÉ

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *INFO*

Par
Sophie LE BARS

Hybrid, logical and linear modelling to predict *in silico* the effect of perturbations on metabolism

Thèse présentée et soutenue à «Nantes», le «9 décembre 2022»
Unité de recherche : LS2N - Laboratoire des sciences et du numérique de Nantes

Rapporteurs avant soutenance :

Alexander BOCKMAYR Professeur, Freie Universität Berlin, Berlin, Allemagne
Hervé ISAMBERT Directeur de Recherche CNRS, Institut Curie, Paris, France

Composition du Jury :

Président : Anne SIEGEL, Directrice de Recherche CNRS, Irisa, Rennes, France
Examinateurs : Alexander BOCKMAYR, Professeur, Freie Universität Berlin, Berlin, Allemagne
Hervé ISAMBERT, Directeur de Recherche CNRS, Institut Curie, Paris, France
Anne SIEGEL, Directrice de Recherche CNRS, Irisa, Rennes, France
Maxime FOLSCHETTE, Maître de conférences, Centrale Lille, Lille, France
Dir. de thèse : Jérémie BOURDON, Professeur, Nantes Université, Nantes, France
Co-enc. de thèse : Carito GUZIOLOWSKI, Maitresse de conférences, Centrale Nantes, Nantes, France

Acknowledgement

First, I would like to thank my supervisor, Carito Guziolowski, and my director, Jérémie Bourdon, who accompanied me during these three years of my doctorate. Despite the complex health situation with meetings more in a video than in person, they have continued to be present and to guide me with a lot of benevolence.

I also want to thank the members of my jury who did me the honour of accepting to judge my thesis. More particularly, the referees who read the integrality of my manuscript thesis.

I want to thank my friends and colleagues who made me laugh and also helped and taught me a lot. Thank you for your good ideas, your advice and your good humour.

I want to thank Albane Lysiak, whom I have known since my master's degree and whom I have had the chance to work with for three more years during my thesis. I also thank Benjamin Churchward, my co-office, for our many - sometimes constructive - conversations, which have helped me progress during the thesis. I have also enjoyed our dog walks with you and my various dogs, enabling me to get some fresh air on the weekends. Thanks to Anna and Marinna for their good mood and their positivism; you are the sunshine of Combi, don't change and keep this attitude, and you are both brilliant researchers. Thank you, Emile, for being yourself, the Einstein of the Combi PhD students, even if your blackboard scares us. I wish all the best to Matthieu Bolteau, who is also supervised by Jérémie and Carito, this year at your side was short and constructive. We exchanged a lot and collaborated on a project together. I congratulate you on your excellent organization, and I do not doubt that you will succeed in your thesis with brio.

I also want to thank the LOGIN association members for all the events organized for the doctoral students, the games and snacks on Friday afternoons, and the exchanges. This association allows a mix between the research teams and to discuss of common points between our different subjects. Therefore, I thank Rémi Garcia, Josselin Enet and Matthieu Bolteau for keeping the association alive with a masterful hand and for growing success. I also thank Rémi for our different exchanges and for teaching me to play Hanobi and love letter. I apologize to Josselin for bothering him several times during the various Login events. You are very good people, and I wish you success in your projects.

I also want to thank my family, who have been very present and have pushed me to the end of the thesis. Thank you, mom, dad, sisters: Coralie, Maud, Caroline and my cousin Anais. I would also like to thank all my animals that I will not list here due to lack of space.

I would also like to thank my friends from the Rennes master's program who shared my adventures via Discord and to whom I wish a lot of success in the future. I hope to continue to see you even afterwards.

Finally, I would like to thank my companion, Grégoire Siekaniec, who has supported me and my moods during these years of doctoral studies. I hope that we will be able to evolve together for a long time to come, both in our private and academic lives. You are a brilliant man, even if you lack confidence in yourself. Thank you also for helping me with my python scripts, managing the animals and making me do sports regularly. I promise to carry a bit more weight on the next trek we make.

Table of Contents

Résumé en français

Les gènes sont des parties de chromosomes qui vont être transcrits en ARNs et, pour certains, traduits en protéines. Les protéines sont constituées d'acides aminés et participent à presque tous les processus biologiques au sein d'un organisme et notamment le métabolisme. Le but premier du métabolisme est de produire l'énergie indispensable à la survie et à la croissance. Les enzymes, une famille particulière de protéines, jouent dans ce cadre un rôle important en catalysant les réactions biochimiques au sein d'un organisme.

Les réactions biochimiques consomment et produisent des composants appelés métabolites. Toutes les réactions biochimiques sont reliées entre elles par leurs métabolites et forment ce que l'on appelle le réseau métabolique.

L'expression génétique établit un pont entre les gènes et leurs transcrits (protéines, ARNm). L'expression génétique est contrôlée par des interactions de régulations entre des gènes et/ou leur transcrits. Le réseau de régulation représente toutes les interactions existantes entre les gènes et leurs produits. L'objectif de ces intéractions de régulation est avant tout de contrôler la production des protéines.

Ces deux réseaux peuvent être représentés sous forme mathématique par un graphe où les composants biologiques sont les nœuds du graphe, et leurs interactions ses arêtes.

Le réseau de régulation et le réseau métabolique sont étroitement liés entre eux. En effet, le réseau de régulation intervient dans la production des enzymes, qui influencent, en les catalysant, les réactions métaboliques et donc la production de métabolites, qui en retour peuvent influencer l'expression des gènes et donc influencer la production d'enzymes.

Les perturbations qui affectent un organisme sont diverses. On peut citer, par exemple, les perturbations liées à une maladie, à un traitement ou à l'environnement. Certains troubles vont avoir un impact sur l'expression des gènes d'un individu. Et donc auront un impact à la fois sur l'expression des gènes et sur le métabolisme puisque ces deux réseaux sont intrinsèquement liés.

La modification de l'expression génétique au cours d'une perturbation peut être observée à l'aide de techniques de séquençage telles que les puces à ADN [1] ou le RNA-Seq [2]. Ces données d'expression sont ensuite disponibles dans des bases de données telles que la base GEO [3].

Cependant, malgré leur imbrication, le réseau de régulation et métabolique ont souvent été étudiés séparément. Cela est dû à la complexité des réseaux biologiques, qui peuvent être composés de plusieurs milliers d'entités et inclure des dizaines de milliers d'interactions. Une autre raison est l'absence d'un lien détaillé entre le réseau de régulation et le réseau métabolique.

Pour certains organismes, les réseaux de régulation et métaboliques sont déjà connus, au moins partiellement, et peuvent être trouvés dans des bases de modèles tels que KEGG [4] ou BIGG [5]. Ces bases permettent de modéliser les deux types de réseaux.

Les réseaux de régulation peuvent être modélisés par des réseaux bayésiens, neuronaux ou logiques (logique booléenne ou floue), ainsi que par des équations différentielles ordinaires (ODE) [6].

Les réseaux métaboliques sont la plupart du temps modélisés en utilsant une approche basé sur les contraintes, tel que l'analyse de l'équilibre des flux (FBA) [7]. La FBA est une méthode mathématique utilisant la programmation linéaire pour étudier le métabolisme. Cette approche permet de trouver une solution optimale basée sur une fonction objective, par exemple la croissance, la production nette d'ATP ou la production d'un métabolite particulier.

Par ailleurs, certains outils de modélisation des réseaux de régulation tel que Iggy [8], utilisent des réseaux de connaissances préalables et des données d'expression génétique, extraites de sources indépendantes, pour comprendre les mécanismes déclenchés par la perturbation d'un système biologique. D'autres outils comme OPT GRAPH [9] proposent des plans d'experimentations *in silico* pour discriminer les modèles de réseaux de régulations. Ici, l'idée est d'utiliser ces approches pour comparer le réseau et les données afin de proposer des prédictions *in silico* qui donnent un nouvel aperçu du système biologique. En raison de la nature incomplète, altérée et bruyante des données biologiques, on s'attend à ce que des comportements incohérents apparaissent lors de la comparaison du réseau et des données. Certains outils se concentrent sur l'identification de ces incohérences [10]. Un comportement incohérent peut être reflété par une interaction manquante, une observation inexacte ou une logique d'interaction mal définie dans le modèle. Dans certains cas, la réparation de ces incohérences est nécessaire pour proposer des prédictions *in silico*.

Des approches de modélisation des réseaux de régulations et métaboliques existent individuellement, mais elles ne relient pas leurs interactions. Dans ce contexte, nous cherchons à étudier l'intégration des réseaux de régulations/métaboliques pour comprendre les mécanismes biologiques en jeu lors d'une perturbation. Nous cherchons également à mod-

éliser des perturbations *in silico*, qui peuvent être la surexpression ou la sous-expression d'un nœud (ou d'un ensemble de nœuds), afin de voir les répercussions que cela peut avoir sur l'organisme et, notamment, si ces nouvelles perturbations peuvent atténuer les effets d'une maladie.

Certaines approches ont déjà été développées pour intégrer les réseaux de régulations et métaboliques. Cette recherche est récente puisque la première approche pour réaliser cette intégration date de 2001 ; il s'agit de la rFBA (regulatory Flux Balance Analysis) : [11]. Dans la rFBA, à chaque intervalle de temps, un état de régulation cohérent avec l'état d'équilibre métabolique est calculé. Ensuite, la FBA est utilisée pour trouver une distribution de flux à l'état d'équilibre pour l'intervalle de temps actuel. Un nouvel état métabolique conduisant à un nouvel état de régulation, le processus est répété jusqu'à ce qu'il n'évolue plus. Cette approche détaillée nécessite un organisme facilement cultivable (*E. coli* par exemple). Une autre approche de 2007, SR-FBA (Steady-state Regulatory Flux Balance Analysis) [12], exprime le réseau de régulation en équations booléennes et le traduit ensuite en équations linéaires, ajoutées comme contraintes dans la FBA. Cette approche nécessite un énorme travail préliminaire pour traduire toutes les équations. Il n'a pu être réalisé que sur un organisme bien connu tel que *E. coli*. Des approches plus récentes, adaptables à des organismes moins connus, existent comme PROM [13] (PRObabilistic regulation of Metabolism). PROM utilise des probabilités pour représenter l'état des gènes qui seront utilisés comme contraintes dans FBA mais nécessite des centaines d'expériences de données d'expression de gènes.

L'un des objectifs de ma thèse est motivé par l'absence d'une approche permettant d'intégrer le réseau de régulation/métabolique de manière plus pratique. La plupart des approches mentionnées sont utilisées sur des organismes bien connus tels que *E. coli* et/ou nécessitent beaucoup de données d'entrée qui peuvent être des paramètres spécifiques ou des milliers de profils d'expression de gènes. C'est la raison pour laquelle j'ai cherché à développer une approche qui intègre le réseau régulation/métabolique sans exiger trop de données d'entrée pour être applicable à un large spectre d'organismes.

Le deuxième objectif de ma thèse est de développer une approche que l'on peut appliquer sur de grands réseaux. En effet, certaines approches existantes ne fonctionnent que sur un petit réseau d'une dizaine de nœuds, alors qu'un réseau de taille réelle comprend environ un millier de nœuds. De plus, avec l'explosion des données biologiques, nous devons traiter un grand nombre de données et en tenir compte lors de la modélisation des réseaux biologiques. Par conséquent, nous voulons une approche qui puisse être appliquée

à l'échelle de l'ensemble du réseau. Pour cela, nous avons dû chercher des stratégies de modélisation adaptées.

Un autre objectif de ma thèse est de modéliser une perturbation existante, comme l'action d'un médicament ou un traitement, sur le réseau intégré et de voir la répercussion sur les différentes couches biologiques. Nous voulons également générer de nouvelles perturbations, qui peuvent être un changement *in silico* de la valeur d'expression dans un seul nœud ou un ensemble de nœuds. Nous voulons voir si cette perturbation a un impact positif ou négatif sur le système.

Par ailleurs, notre objectif est également de disposer d'une approche qui nous permette de comprendre en détail les mécanismes déclenchés par une perturbation afin de localiser, par exemple, quelles réactions métaboliques ou quels gènes ont été impactés.

Dans cette thèse, nous avons cherché à répondre à la question de recherche en présentant deux contributions.

La première contribution est une comparaison d'une approche logique, Iggy, avec une approche bayésienne, Probregnet, pour la modélisation des réseaux de régulation. Elle a été publiée dans la conférence CMSB (Computational Methods in Systems Biology) en 2020 [14]. Notre premier but est de trouver quelles stratégies de modélisation pourraient répondre à nos objectifs en considérant que nous voulons avoir le moins de données d'entrée possible et être capable de modéliser un grand réseau. Nous nous concentrons sur ces deux approches car elles sont considérées comme les plus appropriées pour les études à grande échelle [6]. Dans cette contribution, nous comparons les prédictions des productions des enzymes dans ces modèles suite à une perturbation. En effet, les enzymes sont l'un des liens entre le réseau régulation et le réseau métabolique. Nous avons utilisé les données de deux études précédentes qui se sont concentrées sur la voie de signalisation HIF, connue pour réguler les processus cellulaires dans l'hypoxie et l'angiogenèse et pour jouer un rôle dans les maladies neurodégénératives, en particulier la maladie d'Alzheimer (AD). La première étude a utilisé des ensembles de données sur l'expression génétique dans des tissues extraits de l'hippocampe de 10 patients atteints de la maladie d'Alzheimer et de 13 patients sains, les perturbations et donc les prédictions ont été réalisées *in silico*. La seconde étude a utilisé les données RNA-seq de cellules endothéliales de la veine ombilicale humaine surexprimant *in vitro* la protéine HIF1A. Dans ce cas, l'enzyme a été perturbée expérimentalement, et la prédiction a également été réalisée *in silico*. Nos résultats sur le jeu de données de puces à ADN ont montré que Iggy et Probregnet avaient des prédictions d'enzymes très similaires (73,3% d'accord entre eux) pour la même perturbation.

Sur le second jeu de données, nous avons obtenu des prédictions enzymatiques moins similaires (66,6% d'accord) en utilisant les deux approches de modélisation ; cependant, les prédictions d'Iggy suivaient les résultats mesurés expérimentalement sur l'expression enzymatique. Nous avons conclu que l'approche logique semble être un bon candidat pour atteindre les objectifs de notre thèse. Cependant, l'approche logique présente certaines limites. En effet, ses prédictions ne sont pas facilement quantifiables ce qui rend difficile l'intégration de ces prédictions comme contraintes dans les équations métaboliques. Pour cette raison, nous proposons une nouvelle approche logique basée sur la précédente qui nous permet d'avoir une intégration plus fine.

La deuxième contribution, en cours de publication dans BMC Bioinformatics, a permis d'aborder les limitations mentionnées précédemment. Nous avons développé une nouvelle méthode basée sur Answer Set Programming (ASP), MajS. Iggy s'appuie également sur le langage ASP dans sa mise en œuvre. ASP est un langage de programmation déclaratif adapté pour traiter les problèmes NP de recherche combinatoire.

MajS prend en entrée un réseau de régulation et un ensemble partiel discret d'observations. MajS teste la cohérence entre les données d'entrée, propose des réparations minimales sur le réseau pour établir la cohérence, et enfin calcule des prédictions pondérées et signées sur les espèces du réseau. Nous avons testé MajS en comparant la voie de signalisation HIF-1 avec deux ensembles de données d'expression génétique qui sont les mêmes que dans notre première contribution. Nos résultats montrent que MajS peut prédire 100% des espèces non observées. En comparant MajS avec deux outils similaires, un outil qualitatif, Iggy et un outil quantitatif, Probregnet. Nous avons observé que par rapport à Iggy, MajS propose une meilleure couverture des espèces non observées, est plus sensible aux perturbations du système, et propose des prédictions plus proches des données réelles. En outre, MajS fournit des prédictions discrètes plus raffinées qui sont en accord avec la dynamique proposée par Probregnet. En conclusion, MajS est une nouvelle méthode pour tester la cohérence entre un réseau de régulation et un ensemble de données qui fournit des prédictions sur des espèces non observées dans le réseau. Il fournit des prédictions discrètes à grain fin en sortant le poids du signe prédit comme un élément d'information supplémentaire. La sortie de MajS, grâce à son poids, permet d'envisager une meilleure intégration pour la modélisation des réseaux métaboliques.

Cette thèse est structurée autour de 3 chapitres et d'une conclusion générale du travail de recherche effectué.

Le Chapitre 1 présente l'état de l'art en biologie et en informatique sur les données

biologiques et la modélisation des réseaux.

Le Chapitre 2 est consacré à la première contribution, j'y compare deux approches de modélisation du réseau de régulation ; l'une utilise la programmation logique, l'autre les réseaux bayésiens. Les deux permettent de générer une perturbation *in silico* sur le réseau de régulation. Nous avons cherché à savoir quelle méthode semblait la plus adaptée pour atteindre les objectifs de notre thèse.

Le Chapitre 3 se concentre sur ma deuxième contribution et présente une autre approche logique que nous avons développée, qui permet d'avoir une répercussion plus facilement quantifiable lorsque notre réseau de régulation est perturbé. Le but est de faciliter l'intégration future du réseau de régulation et du réseau métabolique.

Le dernier chapitre est une conclusion globale de tous les chapitres et présente les perspectives.

INTRODUCTION

Context

The genes are parts of chromosomes that will be transcribed into RNA and, for some, translated into proteins. Proteins are made of amino acids and participate in almost all biological processes within an organism, for example, its metabolism. The primary purpose of metabolism is to produce energy essential for survival and growth. The role of certain proteins called enzymes is, in particular, to catalyse biochemical reactions within an organism's metabolism.

Biochemical reactions consume and produce components called metabolites. All biochemical reactions are linked together by their metabolites and form what is called the metabolic network.

The genetic expression bridges genes and their transcripts (proteins, mRNA). Gene expression is controlled by regulatory interactions with other genes and/or transcripts. The regulatory network represents all the existing interactions between genes and their products. The purpose of the regulatory network is, above all, to control the production of proteins.

Both networks can be represented in mathematical form by a graph where the biological components are the nodes of the graph, and their interactions are its edges.

The regulatory and metabolic networks are closely linked together. Indeed, the regulatory network intervenes in the production of enzymes, which influence, by catalysing, the metabolic reactions and, therefore, the production of metabolites, which in return can influence the expression of genes and thus influence the production of enzymes.

The perturbations that affect an organism are diverse. We can cite, for example, perturbations related to a disease, treatment or the environment. Certain disorders will impact an individual's gene expression. And therefore will have an impact on both gene expression and metabolism as the two networks are intrinsically linked.

The modification of gene expression during a disorder can be observed using sequencing techniques such as DNA chips [1] or RNA-Seq [2]. These expression data are available in databases such as the GEO database [3].

However, despite their intertwining, the regulatory and metabolic networks have often been studied separately. This is due to the complexity of biological networks, which can be composed of several thousand entities and include tens of thousands of interactions. Another reason is the lack of a detailed link between the regulatory and metabolic networks.

For some organisms, the regulatory and metabolic networks are already known, at least partially and can be found in model repositories such as KEGG [4] or BIGG [5]. These repositories make it possible to model the two types of networks.

Regulatory networks can be modelled with Bayesian, neural or logic networks (Boolean or fuzzy logic), as well as with ordinary differential equations (ODE) [6].

Metabolic networks are most of the time modelled using Constraint-Based Approach such as the Flux Balance Analysis (FBA) [7]. FBA is a mathematical method using linear programming to study metabolism. This approach allows finding an optimal solution based on an objective function, for example, growth, net ATP production or a particular metabolite production.

Besides, some regulatory network modelling tools such as Iggy [8] use prior knowledge networks and gene expression data, extracted from independent sources, to understand the mechanisms triggered by a perturbation of a biological system. Other tools such as OPT GRAPH, [9] propose *in silico* experimental designs to discriminate regulatory network models. The idea is to use these approaches to compare network and data in order to propose *in silico* predictions which give novel insights on the biological system. Because of the incomplete, altered and noisy nature of biological data, it is expected that inconsistent behaviours appear upon network and data comparison. Some tools focus on the identification of such inconsistencies [10]. An inconsistent behaviour can be reflected by a missing interaction, an inaccurate observation or a wrongly defined logic of the interaction in the model. In some cases automatic repair of such inconsistencies is required to propose *in silico* predictions.

Modelling the regulatory or metabolic networks individually using different approaches is possible, but these approaches do not allow for modelling the existing interactions between both networks. In this context, we aim to study the integration of the regulatory/metabolic networks to understand the biological mechanisms at work during a perturbation. We also seek to model perturbations *in silico*, which can be the over-expression or under-expression of a node (or a set of nodes), in order to see the repercussions that this can have on the organism and, particularly, if these new perturbations can attenuate

the effects of a disease.

Some approaches have already been developed to integrate the regulatory and the metabolic networks. This research is a recent field since the first approach to achieve this integration dates back to 2001; it is rFBA (regulatory Flux Balance Analysis) [11]. In rFBA, at each time interval, a consistent regulatory state with metabolic equilibrium state is calculated. Then, FBA is used to find a steady state flow distribution for the current time interval. A new metabolic state lead to a new state of regulation and the process is repeated until it does not evolve anymore. This detailed approach needs an organism easily cultivable (*E. coli* for example). Another approach from 2007, SR-FBA (Steady-state Regulatory Flux Balance Analysis) [12], expresses the regulatory network in Boolean equations and then translates it into linear equations, added as constraints in the FBA. This approach needs a huge amount of preliminary work to translate all the equations and a well known organism such as *E. coli*. More recent approaches, adaptable to less known organisms, exist such as PROM [13] (PRObabilistic regulation of Metabolism) which appears in 2010. PROM uses probabilities to represent the state of genes that will be used as constraints in FBA but requires hundreds of Microarray data experiments.

Objective

One objective of my thesis is driven by the lack of an approach that allows the integration of the regulatory/metabolic network more conveniently. Most of the approaches mentioned in the previous Context Section are used on well-known organisms such as *E.coli* and require a lot of input data which can be specific parameters or thousands of gene-expression profiles. That is the reason for developing an approach that integrates the regulatory/metabolic network without requiring too much input data to be applicable to a large spectrum of organisms.

The second objective of my thesis is to develop an approach we can apply on large networks. Indeed, some existing approaches work only on a small network composed of tens of nodes, while a real-size network comprises approximately a thousand nodes. Besides, with the explosion of biological data, we must process a lot of data and consider this when modelling biological networks. Therefore, we want an approach that can be applied at the scale of the entire network. For this, we had to look for suitable modelling strategies.

Another objective of my thesis is to model existing perturbation, such as a drug or a treatment, on the integrated network and see the repercussion on the different biological

layers. We also want to generate new perturbations, which can be an *in silico* change of expression value in a single node or a set of nodes. We aim to see if this perturbation positively or negatively impacts the system.

Furthermore, our goal is also to have an approach that allows us to understand in detail the mechanisms triggered by a perturbation in order to locate, for example, which metabolic reactions or genes have been impacted.

From these objectives, my thesis falls within the field of systems biology, which consists of studying and modelling complex biological systems. The central concept behind systems biology is that we should study an organism as a whole rather than individual parts.

Contributions

In this thesis, we sought to answer the research question by presenting two contributions.

The first contribution is a comparison of a logic approach, Iggy, with a Bayesian approach, Probregnet, for regulatory network modelling. It was published in the CMSB (Computational Methods in Systems Biology) conference in 2020 [14]. Our first aim is to find which modelling strategies could answer our objectives considering that we want to have as few input data as possible and be able to model a large network. We focus on these two approaches since these approaches are said to be the most appropriate large-scale approach [6]. In this contribution, we compare the computational predictions of the enzymes in these approaches upon perturbation. Indeed, the enzymes are one of the links between the regulatory and the metabolic network. We used data from two previous studies that focused on the HIF-signaling pathway, known to regulate cellular processes in hypoxia and angiogenesis and to play a role in neurodegenerative diseases, particularly Alzheimer's Disease (AD). The first study used Microarray gene expression datasets extracted from the Hippocampus of 10 AD patients and 13 healthy ones, the perturbation and thus the prediction was made *in silico*. The second one used RNA-seq data from human umbilical vein endothelial cells over-expressing adenovirally HIF1A proteins. Here the enzyme was experimentally perturbed, and the prediction was made *in silico* too. Our results on the Microarray dataset were that Iggy and Probregnet showed very similar (73.3% of agreement) computational enzyme predictions upon the same perturbation. On the second dataset, we obtained slightly different enzyme predictions (66.6% of agreement) using both modelling approaches; however, Iggy's predictions followed exper-

imentally measured results on enzyme expression. We concluded that the logic approach seems to be a good candidate to attain our thesis objectives. However, there are some limitations to the logic approach. Indeed, its predictions are not easy quantifiable which is a problem in providing integration of these predictions as constraints in the metabolic equations. For this reason, we propose a new logical approach based on the previous one which allows us to have a more refined integration.

The second contribution is under revision in BMC Bioinformatics and allowed to address previously mentioned limitations; we developed a new method based on Answer Set Programming (ASP), MajS. Iggy also relies on ASP language in its implementation. ASP is a declarative programming language used to address combinatorial search NP problem.

MajS takes as input a regulatory network and a discrete partial set of observations. MajS tests the consistency between the input data, proposes minimal repairs on the network to establish consistency, and finally computes weighted and signed predictions over the network species. We tested MajS by comparing the HIF-1 signalling pathway with two gene-expression datasets which are the same that in our first contribution. Our results show that MajS can predict 100% of unobserved species. When comparing MajS with two similar tools, a qualitative one, Iggy and a quantitative one, Probregnet, we observed that compared with Iggy, MajS proposes a better coverage of the unobserved species, is more sensitive to system perturbations, and proposes predictions closer to real data. Besides, compared to Iggy, MajS provides more refined discrete predictions that agree with the dynamic proposed by Probregnet. To conclude, MajS is a new method to test the consistency between a regulatory network and a dataset that provides computational predictions on unobserved network species. It provides fine-grained discrete predictions by outputting the weight of the predicted sign as a piece of additional information. MajS' output, thanks to its weight, could easily be integrated with metabolic network modelling.

Outline of the thesis

This thesis is structured around 3 chapters and an overall conclusion of the research work carried out.

Chapter 1 is state-of-the-art biology and computer science on biological data and network modelling.

Chapter 2 is devoted to the first contribution, here I compare two approaches to

model the regulatory network; one uses logic programming, and the other uses Bayesian networks. Both make it possible to generate an *in silico* perturbation on the regulatory network. We aimed to know which method seemed the most suitable to achieve our thesis objectives.

Chapter 3 is focused on my second contribution and presents another logical approach we developed, which makes it possible to have a more easily quantifiable repercussion when our regulatory network is perturbed. The goal is to facilitate the future integration of the regulatory network and the metabolic network.

The last chapter is a global conclusion of all the chapters and presents the perspectives.

Chapter 1

State of the art

Summary of chapter 1

This chapter will provide all the basics, from a biological and modelisation point of view concepts required by the thesis work. It will first introduce all the biological techniques used to generate the datasets referred to in this thesis. The second and third parts will respectively present the notions of biological regulatory and metabolic networks, together with their specificities and approaches to model them. The final part will discuss the existing tools to model the integration between regulatory and metabolic networks.

1.1 Biological datasets

This section presents some important biological techniques to produce some biological data of interest, such as gene expression datasets. Then, we show how this biological data are used to deal with gene regulatory network and metabolic network and allow the integration between them (see Section 1.6).

Introduction to some biological notions

— The genetic information of all living beings is contained in their genome. A genome is composed of one or more chromosomes, depending on the species, made of DNA (DeoxyriboNucleic Acid).

— DNA is a long molecule composed of four nucleotide bases: Adenine (A), Thymine (T), Cytosine (C), and Guanine (G). These bases can bind together by pair (A - T and G - C, see Figure 1.1b) and are arranged in the form of a double helix. The size of a DNA molecule in *Human* is of the order of several tens of millions of bases.

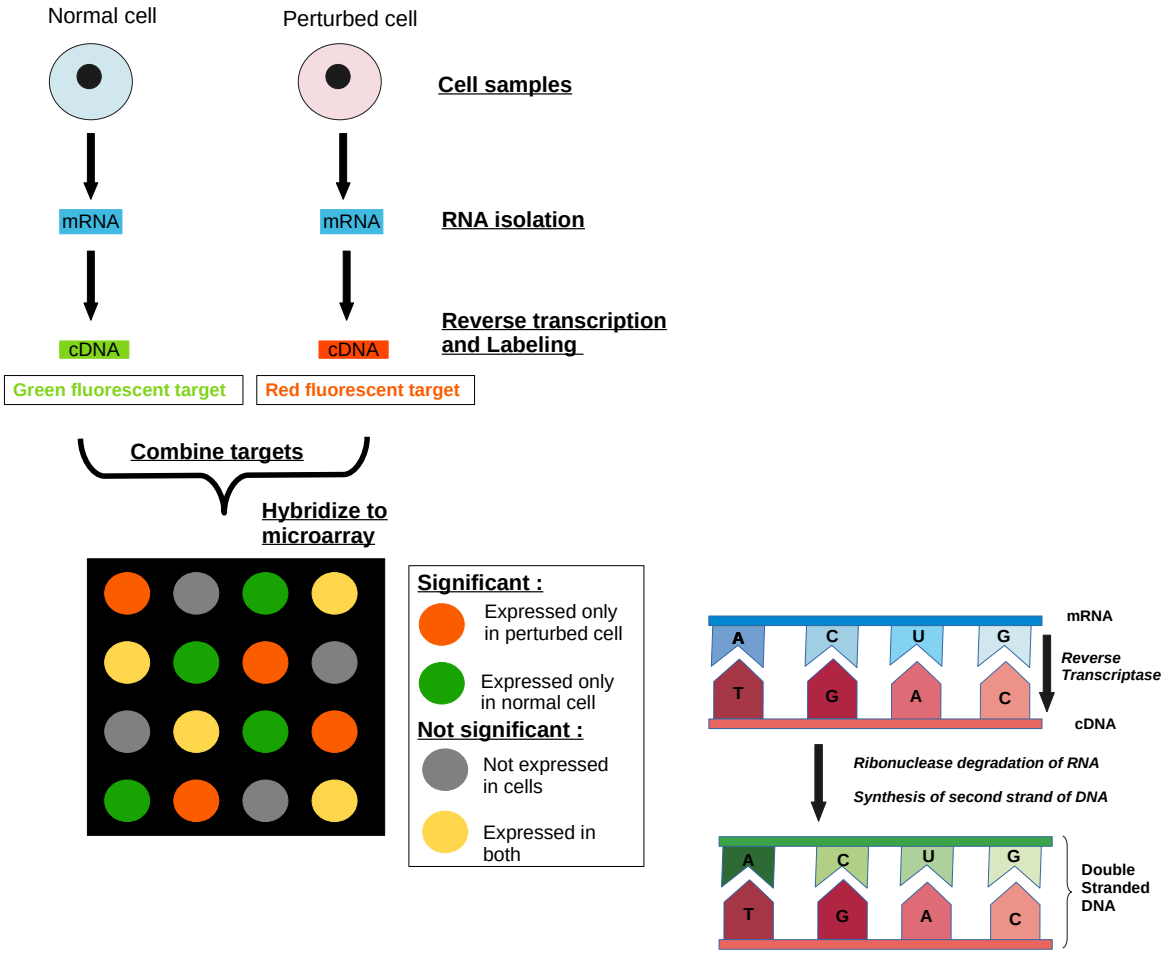
— A gene is a small section of DNA that codes for an RNA molecule also called the gene's transcript. In *Human*, genes vary in size between a few hundred and more than two million nucleotide bases.
— RNA, RiboNucleic Acid, is composed as DNA of four nucleotide bases. The two main differences with DNA are: (1) the Uracil replaces the Thymine, $T \rightarrow U$ (2) RNA is single-stranded whereas DNA is double-stranded, as shown in Figure 1.1b. There are different types of transcripts, such as mRNAs, non-coding RNAs, and small RNAs. Especially, mRNA is an intermediate molecule that carries the genetic information for protein synthesis.
— mRNAs can be translated into proteins, molecules composed of amino acids. Proteins are made from amino acids which are chained together. For eukaryotes species, there are 20 different amino acids, and a protein gathers 450 amino acids on average. Proteins participate in almost all biological processes inside an organism [15].

DNA Microarray Sequencing

DNA microarray, also called DNA chip, allows the sequencing i.e. the process of determining the DNA sequence, of thousands of genes simultaneously [1] [16]. This technique was first mentioned in 1995 by Murray *et al.* [17]. DNA microarray tools are mainly used to detect gene expression variation between a normal and perturbed condition. Indeed this variation of expression can affect the production of proteins and thus result in a disease of the concerned organism.

As part of the study of a disease, the operating principle of a DNA chip is described in Figure 1.1a. First normal and perturbed cells are collected from an organism, and mRNA is isolated. Afterwards, this mRNA is converted by using an enzyme into a more stable form, cDNA, as mRNA degraded quickly; the link between both is shown in Figure 1.1b. Eventually, cDNA is labelled using fluorochrome dyes Cy3 (green, used for the normal cell) and Cy5 (red, used for the perturbed cell). The cDNA is inserted into a chip and hybridise -or bind- with the synthetic cDNA of the chip. A chip is composed of a collection of DNA spots attached to a solid surface. These spots contained specific DNA with and without mutation. The cDNA will hybrid with one spot or another depending on their sequence similarity. When a cDNA binds to a spot, this activates the attached fluorochrome. Thus, we can see which are the genes expressed in each condition [16].

The raw output of DNA microarray is raw light intensities. These are then converted



(a) (b)

Figure 1.1 – (a) **DNA microarray principle** (b) **The link between cDNA, mRNA and double stranded DNA**

into gene expression levels after some preprocessing steps, i.e., applying a background correction, normalising, and summarising the results.

RNA-Seq

RNA-Seq is one of the next-generation sequencing (NGS) technologies that revolutionized transcriptome analysis [2] [18]. RNA-Seq allows the analysis of the transcriptome, i.e. all expressed transcripts (RNAs) of an organism at a given time and under given conditions.

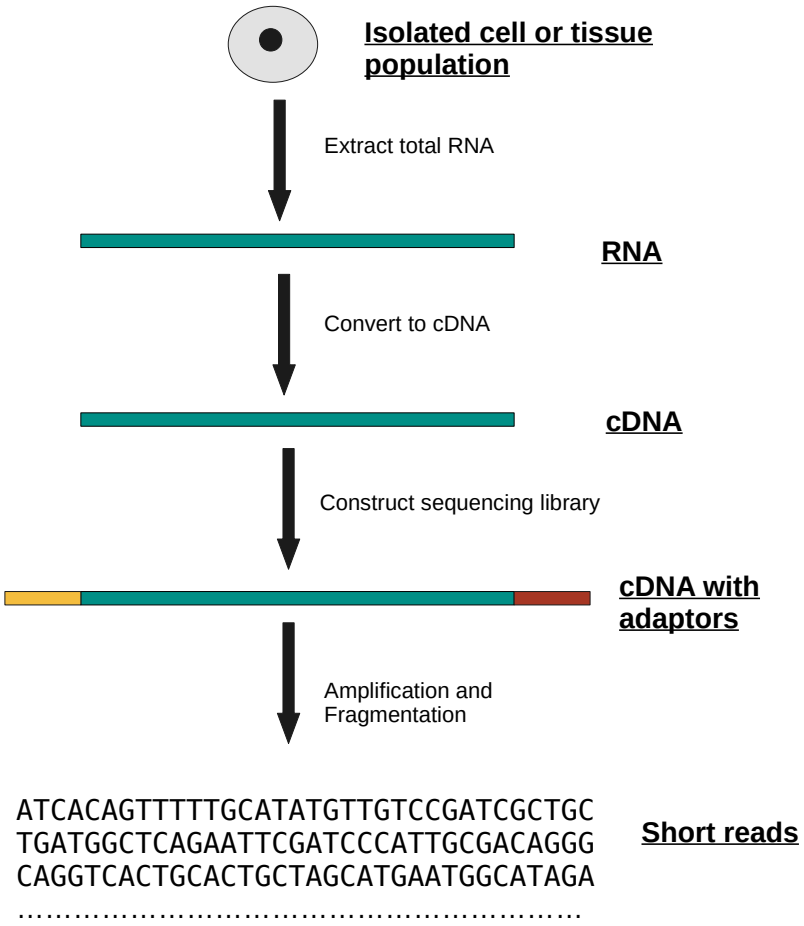


Figure 1.2 – **RNA-Seq general principle**

The RNA-Seq principle with a standard NGS platform is summarized in Figure 1.2. We present here the most widely used NGS sequencing system, Illumina [19]. The first step is RNA extraction from a biological sample as a cell or tissue. Then, RNA is fragmented and converted into cDNA, and adaptors are added. Afterwards, the cDNA fragments are amplificated via a bridge PCR.

In a bridge PCR, the cDNA fragments will be attached to a flow cell via their adaptors in the form of a bridge. Then a complementary strand is synthesized using a polymerase, an enzyme that synthesizes a chain of nucleic acids (in the same way as Figure 1.1b). A denaturation step will then separate the two strands. This operation will repeat and form a cluster of identical strands.

The sequencing first cycle begins in Illumina by adding fluorescent-labelled nucleotides in the flow cell; after laser excitation, the emitted fluorescent for each cluster of identical

strands is captured. The colour of the emitted fluorescent (4 shades in total) allows the identification of the nucleotide base for each cluster. This cycle will repeat n times by adding new fluorescent nucleotides and exciting with a laser each time. n represents the final size of the estimated read; this repetition can be parameterized and has to be put in correlation with the size of the fragmented cDNA. In the end, short reads (a small portion of DNA) are obtained. The short reads are used to estimate the abundance of transcripts for each gene; it is then possible to evaluate the gene expression.

Recently developed, the RNA-Seq technique; appeared in the mids 2000 and has many advantages compared to other approaches such as DNA microarray. Indeed, DNA microarray has numerous limitations that do not appear in RNA-Seq [16]. First, DNA microarray will give an indirect measure of the concentration of the gene, which is less precise than RNA-Seq. Second, DNA microarray is not as specific as RNA-Seq; it may detect a given gene as well as its homologs. The third limitation is that DNA microarray can only see the sequence that the array was designed to detect. It signifies that if a gene has not been annotated yet, it won't be present in the array. In the end, we also cannot study non-coding RNA with DNA microarray. However, DNA microarray is suitable for routine analyses because it is fast and inexpensive and for tests where we expect to have results on one or more genes. In contrast, RNAseq is more exhaustive and better for research-type analyses where we do not know the desired results. Nonetheless, there are some limitations to RNA-seq: if we do short-read sequencing, there may be biases and imperfections during the preparation of the sequencing library or the assembly. If we do long-read sequencing, it is more expensive, and there are more sequencing error rates. However, RNA-Seq has allowed great advances in the characterisation and quantification of the transcriptome.

Both techniques are used to generate gene expression datasets and estimate the impact of a perturbation on the gene expression level. To understand the effect on a larger scale, we need to study all the interactions between genes and their products, using biological networks such as gene regulatory networks. Gene expression datasets are available in a lot of biological databases. GEO database [3] is one of the most extensive gene expression databases where more than 127 450 organisms are listed for different experimental conditions using both presented techniques.

1.2 Biological networks

Biological networks represent complex biological systems (e.g., cells, tissues, whole organisms) [20]. Biological networks can be based on literature or based on data, it relies on biological interactions and allows to represent different entities interacting together. Biological networks allow us to understand the mechanisms triggered by a perturbation such as a disease, drugs, or environment. Omics data, which represent molecules of the same type from a biological sample [21] can be used to understand biological interactions. There are different types of omics data: genomics data, which englobe the complete set of genetic information in an organism, or transcriptomic data, which represents all the transcripts (RNAs) of an organism. Data acquisition is a crucial step in building biological networks. The two sequencing techniques presented above allow us to obtain such data. Moreover, there are different types of biological networks, such as:

— the Protein Interaction Network, which represents physical contact between proteins in an organism.
— The Gene Regulatory Network represents a collection of genes interacting together.
— The Metabolic Network represents the connection between biochemical reactions.
— The Signalling Network represents the signalling pathways interacting together.

In this manuscript thesis, we will focus on gene regulatory and metabolic networks.

1.2.1 Gene regulatory network

Gene regulation aims at controlling the synthesis of gene products (mainly mRNAs or proteins) in cells. A functional gene regulation leads to a distinct phenotype in a biological system and allows stability. When misregulation occurs, it is generally associated with a disease. Moreover, gene regulation is essential for the adaptability of an organism allowing it to face different environmental changes [22]. The gene regulatory network (GRN) is a system biology object to understand the mechanisms that control a cell's or organism's response to stimuli.

Some of the gene products, the transcription factors (TFs), are key players in the regulation mechanisms. Indeed, TFs are DNA-binding proteins that modulate one of the first gene expression controls. A TF can bind to a specific region of a target DNA, called the cys-regulatory region of genes. The TF would then inhibit or enhance the associated gene(s) expression and thus impact the RNA production of this gene. This information will be transmitted downstream in the network; indeed, among RNAs, some may encode

for TFs whose expression will thus also be perturbed, which will have repercussions on their target gene. The regulation mechanism is a complex system composed of multi-level feedback from genes and TFs. The whole system determines how an individual would react to stimuli.

The GRN structure is as follows, genes and their products are represented by nodes inside the network and the interactions between the nodes are represnted by edges, which can be activation or inhibition. The nodes in GRNs are TFs, genes, mRNAs, proteins. These networks are inferred from literature (biological expertise) or from experiments (biological data) to understand the interactions between TFs and genes or find new regulatory elements (TF-gene binding). Databases such as Uniprot and JASPAR gather eukaryotic transcription factors and their binding targets. Other databases related to GRN can be found such as RegNetwork [8], RegulonDB [23], TRANSFAC [24] and GRNdb [9].

1.2.2 Metabolic network

Cellular metabolism is generating an energy source and essential products (e.g. vitamines) that allows cell growth and survival. The energy produced by a cell is in the form of a molecule called adenosine triphosphate (ATP). Inside the cell, biochemical reactions are catalyzed by enzymes, a class of proteins. A biochemical reaction transforms one or multiple molecules called reactants into one or multiple different molecules called products, all these molecules (reactants and products) are called metabolites. Some reactions that share metabolites form what is called a metabolic pathway. The reactions inside a metabolic pathway are linked as metabolites produced by a reaction are then consumed by another. There are two types of metabolic pathways: the one synthesizing molecules using energy (anabolic pathways) or releasing energy by consuming molecules (catabolic pathways). Metabolic pathways are strongly interconnected. Indeed, the same metabolites are found in different biochemical reactions as reactants or products, and an enzyme can also catalyze different reactions. The metabolic enzymes represent an important part of the gene products; for example, in *S. cervisiae*, a model organism, the percentage of genes that code for enzymes is 14 %. This is an illustration of the existing connection between the different biological layers.

A metabolic network represents the connection that exists between the various metabolic pathways [22], and can be found in databases such as KEGG [25] and BIGG [5].

The metabolic and regulatory networks are tightly connected via different entities, which are enzymes and metabolites (see Section 1.6).

1.3 Using Answer set programming to model biological networks

This section introduces the Answer Set Programming (ASP) language and presents examples of some ASP applications to model biological systems.

Bases of Answer set programming

Answer set programming (ASP) is a declarative programming language used to address combinatorial search NP problems. In a declarative programming language, rather than figuring out how to solve the problem (imperative programming), we try to define what the problem is; see Figure 1.3 for an illustration.

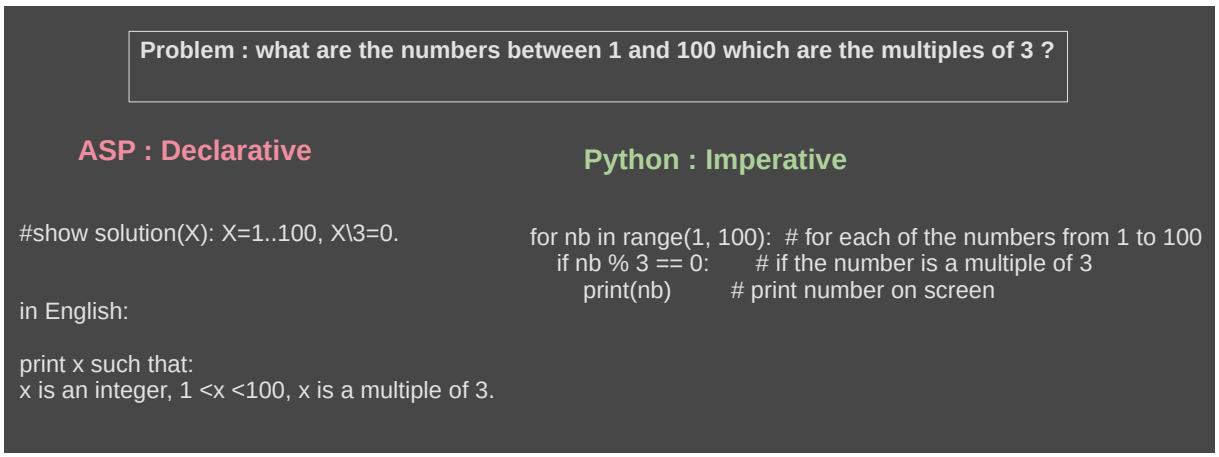


Figure 1.3 – Comparison of the declarative and imperative language to resolve the same problem

ASP can define logic programming rules, expressed using first-order logic, within a discrete domain and find stable Herbrand models which satisfy these rules [26]. Logic programming is a sub-paradigm of declarative programming, in which statements express facts and rules following formal logic. A logic program is composed of the following ingredients:

— Generate: rules to generate the set of potential solutions. (grounding)
— Test: rules to trim the set of potential solutions, eliminating unwanted ones. (solving)
— Define: (optional) rules to define auxiliary predicates.

Figure 1.4, illustrates the ASP resolution process. The ASP version used in this thesis is the implementation done by POTASSCO (Potsdam Answer Set Solving Collection). More precisely, we used Clingo [1], an ASP tool combining Gringo, a grounder software and Clasp, a solver. Gringo translates the program choice rules into a grounding program composed of models with only constants and no variables. Clasp takes as input the grounding program and filters the models by applying the constraint rules inside the logic program. In the end, if the solver finds a stable model which satisfies all the constraints, it will output this solution. Else, the solver outputs unsatisfiable.

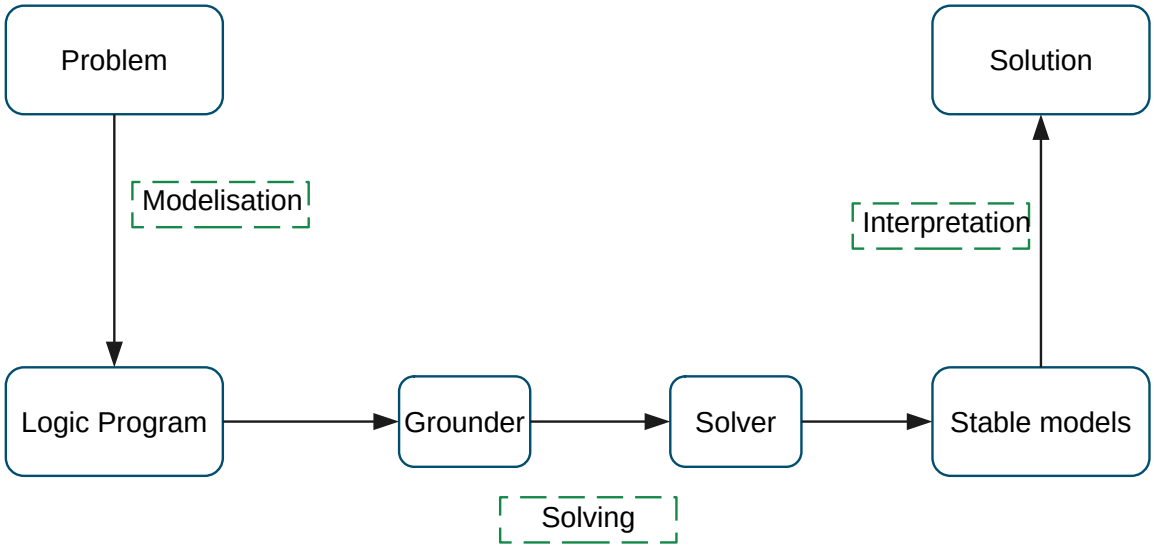


Figure 1.4 – ASP resolution process

For some problems, optimisation constructs are defined to find optimal solutions in a solution space. Thanks to ASP, it is also possible to handle intersection, union, enumeration and optimisation of models. In addition, unlike other declarative approaches (e.g. Prolog), ASP allows one to work with negation by default: a predicate is false as long as no other rules in the program allow to say that it is true.

ASP language notions

ASP uses the same semantics as logic programming. Indeed, the facts and rules in ASP are expressed using a formal logic. The relation between two objects is represented

1. `https://github.com/potassco/clingo`

with *predicates*. For example, to represent the relation between a father and his son, we can use a predicate *father*: All codes below can be tested here [2].

```
%comment in ASP
father(darkvador,luke).%darkvador is the father of luke
villain(darkvador).%darkvador is a villain
good(luke).%luke is a good guy
```

villain and *good* are also predicate. *darkador* and *luke* are constant term. All constants in ASP begin with a minuscule. In contrast, variables are represented in capital letters in ASP. For example, X in the figure 1.3 is a variable that can take a value between 1 and 100.

An *atom* is a term or a predicate which is *grounded*; it means that this expression does not contain any variable. An atom is a true fact. For example, ASP considers the predicate *villain* as an atom, so the fact that darkvador is a villain is true. The predicate *father* is of *arity* 2 which signifies that two terms are concerned whereas predicates *villain* and *good* are of arity 1. The predicate is noted as <name_of_predicate>/degree_of_arity. In our case, the predicates are father/2, villain/1 and good/1.

The rules are written as logical clauses and represent a relationship of causality in the form *head:- body.* The interpretation of a rule is "the head is true if and only if the body is true". Using rules, it is possible to define whether an atom is true under certain conditions. If a rule is composed of only a head and no body then this rule is perceived as a true fact (*head.*). An example of logical rules:

```
villain(darkvador).villain(joker).%same as villain(darvador;joker).
good(luke).good(batman).
 ennemy(X,Y) :- villain(X); good(Y).
```

The translation of this rule is "if X is a villain and Y is good, so they are enemies".

Additionally, we can also make choices in ASP using this formulation:

```
 1{villain(darkvador;thanos;joker)}1.
```

Here we ask ASP to choose a villain among the 3 given villains. ASP will therefore offer us 3 different solutions, each representing 1 villain. We can constrain this choice and decide to have a villain who is "not thanos", for example, by adding a negation expressed like this:

```
 not villain(thanos).
```

2. https://potassco.org/clingo/run/

ASP will propose only 2 solutions by excluding Thanos from the possible solutions. Other operations are also possible in ASP, such as counts. For example:

```
nb(N):-N=#count{villain(darkvador;thanos;joker)}.
```

This will return one solution, which is nb(3).

ASP possesses optimizations operation, which allows to obtain optimal solutions noted #maximaze or #minimize. Example:

```
1 { nb(1..100) } 1.%choose a number between 1 and 100.
#maximize{N:nb(N)}.%Maximize N which belong to the search space obtain
    with predicate nb.
```

The optimal solution returned is nb(100) which is the maximal number between 1 and 100.

Example of using ASP to model biological systems

ASP has proven to be an efficient language in systems biology, it was used for analyzing metabolic, signalling and regulatory networks but also for consistency checking, diagnosis and repair of biological data and models. For example, meneco is an ASP tool that reasons on metabolic networks. meneco checks if a metabolic network draft can produce observed metabolites and automatically complete the draft until the observed metabolites are produced [27]. For ASP tools on signalling networks, we can mention caspo [28] that infers logical networks from experimental data and designs new experiments to reduce uncertainty and look for strategies to control the biological system behaviour. Finally, Iggy which is used in my thesis is one of the examples of ASP applications to check the consistency of regulatory networks and observe system behaviour; see Section 1.4.1 for details.

1.4 Gene regulatory network modelling

A gene regulatory network is composed of macromolecules; the majority of them are proteins that interact to regulate the level of expression of genes.

For many organisms, regulatory networks are already known, at least partially, and can be found in databases such as KEGG [4] or BIGG model repositories [5].

The aim of modelling regulatory networks is to explore the relationship between genes and determine the dynamics of the genes inside an organism. By understanding the dynamics of this network, we can shed light on the mechanisms triggered by a perturbation

on an organism. There are a lot of different approaches; we are providing a limited list. We will focus on Logic and Bayesian networks since they seem to be the most appropriate, large-scale approaches for studying a whole organism [6]. These two approaches can be used on bigger networks, knowing that an organism is composed of thousands of genes and hundreds of transcription factors (TFs). As an example, the bacteria *E. coli* comprises approximately 4000 genes, and 200 TFs [29].

1.4.1 Logic network

Logic-based network models were among the first approaches used to model regulatory networks. They were introduced by Kauffman in 1969 [30]. These approaches present a good compromise between complexity and precision [31]. They need as input data prior knowledge of the structure of the biological network but compared to other approaches; they do not need so many parameters (e.g., kinetic parameters). There is a wide range of logic-based models, such as Boolean network models, Generalized Logical Networks, and probabilistic Boolean networks.

Boolean network

A node in the boolean network represents each component of the regulatory network. Each directed edge between two nodes represents their regulatory interactions.

A node of the network, gene or protein, is modeled as a binary device (ON-OFF). ON/1 when it is active OFF/0 when it is inactive. So a Boolean network composed of n nodes may have 2^n states. Boolean networks contain a set of nodes, $G = \{g_1, g_2, ..., g_n\}$. The state of a node g_i is modeled by a Boolean function f_i that represents the regulatory dependencies with its direct predecessors in the boolean network. The set of Boolean functions for all the nodes is denoted as $F = \{f_1, f_2, ..., f_n\}$. When Kauffman introduced Boolean networks in 1969 it was stipulated that the time of this model is discrete so at each time t there is a new state of the system which is updated [30]. A state of the system corresponds to an ON/OFF assignment for each node in the network. The transition of a node g_i from time point t to time point $t+1$, $g_i(t) \rightarrow g_i(t+1)$, is represented by a Boolean function: $g_i(t+1) = f_i[P(g_i)(t)]$ where $P(g_i)(t)$ represents the values of the parents of node g_i at the current time point t. This basic Boolean network can update from time point t to time point $t+1$ by using synchronous update. A synchronous update signifies that all Boolean functions are applied to transit for each node from $t \rightarrow t+1$ at the

same time [32]. Other updating schemes exist, such as asynchronous update. Therefore, the dynamics of the Boolean network are deterministic as each state of the system has one successor; only one new system state is possible after an update of the system.

Different approaches exist to construct Boolean networks, such as literature-based methods that require user knowledge and literature research [33]. When experimental data are available, we can use data-driven methods that will allow us to infer Boolean functions [34] using gene expression data.

Boolean network modelling allows studying the regulatory network dynamics with limited knowledge. Although it simplifies biochemical processes highly, it is still helpful to discover new biological knowledge [35]. Boolean networks show only two states (levels of species expression), ON or OFF. Some mechanisms based on different expression levels cannot be modeled using this approach. Boolean networks can be extended to the Generalized Logical Networks (GLN) to solve this problem.

Sign-consistency approach

The sign consistency approach tests the consistency between an interaction graph (IG) and a list of partial discrete observations of this graph nodes derived from experimental datasets. The IG is a signed directed graph, where the edges are signed as "+" or "-" and directed so that $i \rightarrow j$ means species i influences species j. The list of discrete observations is composed of discrete ("+", over-expressed; "−", under-expressed; "0", no-change) changes associated with some nodes. This change represents the differential expression of a gene between two system conditions (for example, normal and perturbed). Given a *sign consistency rule*, a graph is said to be consistent with respect to a list of discrete observations if the change of a node agrees with the network topology and the list of given observations. In case of inconsistency, the modelling framework proposes artificial repairs allowing to establish consistency. After consistency is established, the modelling agrees on new discrete changes on some initially unobserved species; these agreements are called predictions.

Iggy framework

Iggy [8] [3] is a framework based on ASP that uses sign consistency modelling. The observations represent a change of expression between two conditions assigned to some

3. `http://bioasp.github.io/iggy/`

graph nodes. It automatically detects inconsistencies between graph and observations, applies minimal repairs to restore consistency, and predicts the sign of unobserved nodes by applying the following logical rules:

1. The observations must keep their initial sign.
2. The "+" or "-" sign for each signed node n must be justified by at least one of its received signed influences. An influence from node p to n, is the product between the (p, n) edge sign and p's sign.
3. Each node signed as "0" must have only one influence signed as "0" or at least one "+" and one "−" influence.

Iggy proposes a set of consistent models. Then, Iggy summarises all consistent models in a step called Projection. Iggy has 6 different levels of predictions which are estimated after the Projection step: "-", "notPlus", "0", "notMinus", "+", "CHANGE".

"-", "0", "+" are strong predictions as the node is always predicted with the same sign in all consistent models.

"notPlus", "notMinus", "CHANGE" are weak predictions: a node can be predicted with different signs across all consistent models ("notPlus": {"−", "0"}; "notMinus": {"+", "0"}; "CHANGE": {" + "; " − "}).

Another output of this sign-consistency approach is a list of inconsistent nodes in case of incompatibility between a graph and an observation list. Iggy needs to fix these conflicts to be able to make the prediction. It proposes a repair by adding artificial interactions in the network and will propose a minimal correction set (MCoS) of added interactions. In some cases, if multiple repairs are possible, Iggy will compute them all, and the final set of predicted nodes is the union of predictions obtained after each repair. Iggy allows solving this combinatorial problem by using a solver Clasp [4]. Iggy's application on a toy example is given in the following paragraph.

Iggy application on a toy example

This toy example is composed of 10 nodes, 7 activation edges, and 1 inhibition edge ($E \dashv D$). In Figure 1.5 we illustrate how Iggy proceeds when comparing this toy IG with one dataset of observations. First, Iggy recovers consistency by adding only one artificial influence (art) on node I. Then, it predicts values over nodes D, E, and G, which are

4. Last version of Iggy, 2.2.0 relies on Clasp 3.3.6.

unobserved. Figure 1.5 shows the prediction of Iggy for the three unobserved nodes and the repaired node, I. This toy example outputs three optimal answer sets: *Solutions 1*, *2* and *3*. Focusing on node E, we observe that in the answer set *Solution 1*, the predicted sign is "-", in *Solution 2*, the predicted sign is "+" and in *Solution 3*, the predicted sign is "0". Node D and G are always set to "+" as explained by the sign of their received influence (see rule 2). To illustrate how projection works with Iggy, let us focus on nodes D and E. For node D, the sign across all optimal solutions is "+" so the sign given by the projection is of "+" (Figure 1.5 (b), column *Prediction*). For node E, the sign varies between $-$, $+$ and 0 across all optimal solutions; in this case, Iggy cannot give a prediction. Finally, in Figure 1.5 (b), we see that the sign of node I is the same as its observed sign. Indeed, the added artificial influences allow the node to keep the observed sign despite the inconsistency. The inconsistency is explained by J, which is the only predecessor of I and activates it. Thus, J and I should have the same sign in consistent local behaviour. To guarantee a global consistency of the whole network, node I had to be repaired.

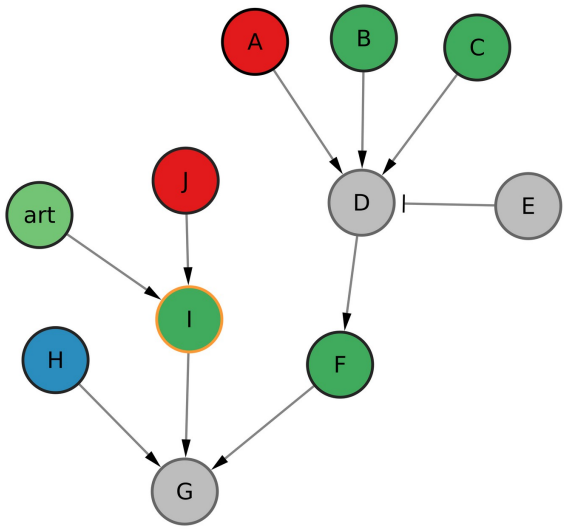
1.4.2 Bayesian Network

Bayesian networks are also called probabilistic directed acyclic models [36]. Bayesian networks allow a representation of conditional dependencies between random variables as represented in Figure 1.6. Bayesian networks rely on the Bayes theorem, which describes the probability of an event based on prior knowledge of conditions that might be related to the event. This theorem uses conditional probabilities, which measure an event's probability, given that another event has already occurred.

The nodes are random variables $X = (X_i)$, $i = 1, .., n$ where n represents the number of components inside the network. The edges represent the conditional dependencies. $p(X_i|a_i)$ is the probability for the variable X_i conditioned by the set of its parents in the graph, a_i. The joint distribution of all variables is:

$$P(X_1, X_2, ..X_n) = \Pi_{i=1}^{n} p(X_i|a_i)$$

By taking a gene expression dataset D, a Bayesian approach will give a quantitative assessment if a directed acyclic graph noted G, will produce such data. The main limitation is that learning G using D is an NP-hard problem as the number of possible graph structures increase exponentially as the size of the Bayesian network increases [38]. Therefore, heuristic searches are used to find an approximation of this problem solution



(a)

Node	Solution 1	Solution 2	Solution 3	Prediction
D	+	+	+	+
E	-	+	0	NA
G	+	+	+	+
I	+	+	+	+

(b)

Figure 1.5 – **Toy case study Iggy**. (a) Toy network with 7 nodes that are initially observed and 3 unobserved nodes. The *I* node is marked as inconsistent. The colours for observed nodes is: "+" (green) if the node is over-expressed, "-" (red) if under-expressed, and "0" (blue) if there is no change of expression between the two conditions. The unobserved nodes are in grey (b) Iggy predictions on toy network example. Unobserved nodes (grey) are predicted by Iggy with a sign. The orange node is repaired by adding one artificial influence. Columns Solution 1, 2 and 3 represent sign in optimal answer sets for unobserved and repaired nodes. Column Prediction is summarizing all Solution columns.

[39]. The Bayesian network formalism does not allow to take into account any form of a loop that can appear in a Biological system and does not consider the dynamic process of gene regulation. For that purpose, dynamic boolean networks have been introduced.

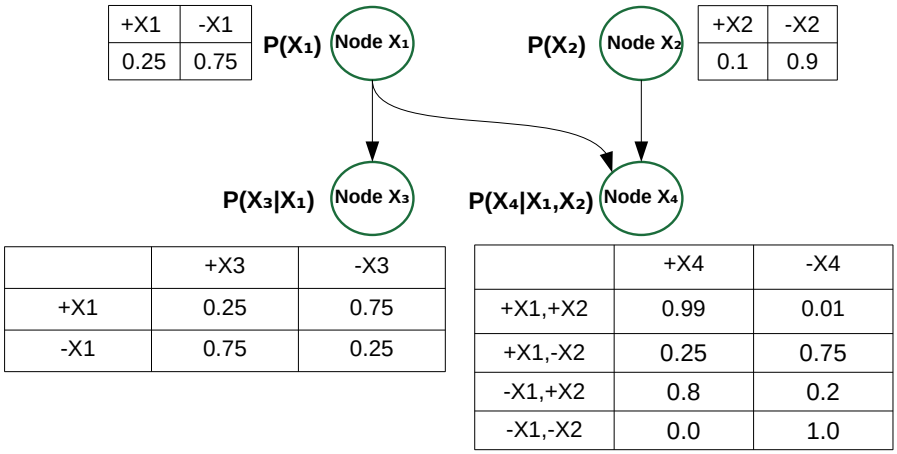


Figure 1.6 – **Representation of a small Bayesian Network with fictitious number** network inspired by [37]. The fictitious number represents the probability that a gene (X) is active (+) or inactive (-) depending on the state of its predecessors in the graph.

1.4.3 Other existing methods

Among other existing formalisms to model regulatory networks we can cite neuronal networks [40], state-space model [41], differential equation model (ODE) [40] and relevance model [42]. These formalism are based on reverse engineering using data to model the regulatory network. However, these methods use either temporal data or require a lot of additional parameters not always available. As we are working with non-temporal data and want as few parameters as possible to have a generalizable method for most organisms we did not present in detail these methods.

1.5 Metabolic network modelling

A metabolic network is composed of biochemical reactions that produce and consume metabolites. These biochemical reactions are catalyzed by enzymes. The principal aim of the metabolic network is to produce biomass [43].

There are multiple goals fulfilled by modelling metabolism; some of them are: enhancing bioprocess performance, a better comprehension of cell biology, drug target discovery, and metabolic therapy. In this section, we present an introduction to one approach com-

monly used for modelling metabolic networks. For a more detailed introduction, please refer to [7].

1.5.1 Constraint-Based modelling and Flux Balance Analysis

Flux Balance Analysis (FBA) [44] is a mathematical method using linear programming to simulate metabolism.

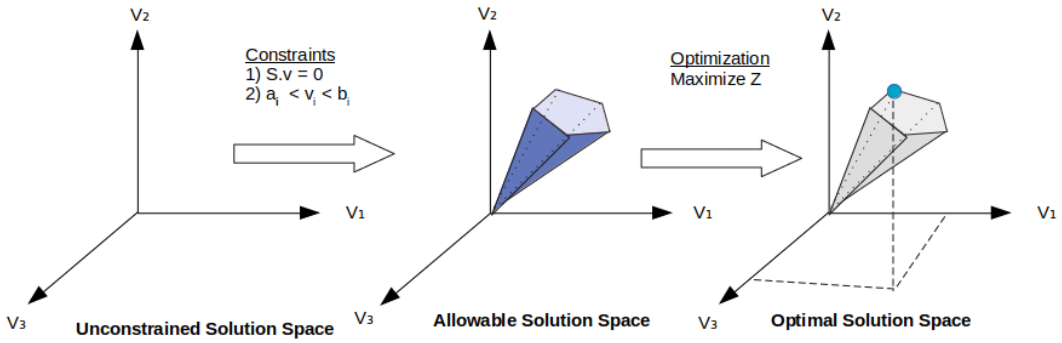


Figure 1.7 – **Principle of FBA in three step** inspired by [44]

In the first step, A metabolic network is defined by (m,n,S) where m represents the number of compounds in the metabolic reactions, such as metabolites and enzymes. n represents the number of biochemical reactions. S represents the metabolic reactions as a numerical matrix called the stoichiometric matrix of dimension $m * n$. The cells of this matrix are the stoichiometric coefficients of each compound for each reaction. If for a cell c_{ij}, the stoichiometric coefficients is of 0 it signifies that the compound i does not participate in the reaction j. A positive coefficient signifies that the reaction produces the compound, and a negative one signifies that it is consumed. A vector represents the flux through all the reactions noted v, which has a length of n. In FBA, we assume that the metabolism is at a steady state. This assumption is defined by a constraint called **mass balance constraint**, $S.v = 0$. Another constraint is added, named the **capacity constraint**, which stipulates that reaction fluxes are between a lower and an upper bound: $a_i < v_i < b_i$. a_i is for the lower bound, b_i, the upper bound, and v_i for the i-th reaction of the metabolic system. These constraints can be found in Figure 1.7 and define an allowable solution space in the form of a polyhedral cone (blue in the figure). This solution space represents all the fluxes acquired for the metabolic reactions

subject to the two constraints. FBA then searches to optimise (maximize or minimize) an objective function Z, that represents, for example, the growth rate or the maximum ATP production for an organism. By applying the objective function Z, we find an optimal solution, in general, on a vertex of the cone. This optimal solution corresponds to the optimal distribution of fluxes to achieve the objective function.

1.5.2 Other existing approaches

Other approaches exist to model the metabolic network using, for example, thermodynamic-based constraints or kinetic parameters. All these approaches are detailed in [45]. They aim to be closer to the biological reality, but the disadvantage is that they need to have more input data than FBA.

1.6 Integration between regulatory and metabolic networks

For many years the regulatory and metabolic networks have been studied independently, but they are deeply related together, as illustrated in Figure 1.8. Indeed, the enzymes produced by the regulatory network catalyze some reactions in the metabolic network. Besides, metabolites produced by the biochemical reactions can regulate the expression of transcription factors inside the regulatory network. The intertwining of these two networks could improve our global comprehension of the effect of a perturbation on an organism; this is the crucial idea of this thesis. This section makes a non-exhaustive review of the tools that already make it possible to integrate these two biological networks.

1.6.1 Approaches translating regulatory effects into Boolean rules

rFBA, regulatory Flux Balance Analysis

The first approach rfBA was used on *E. coli* in 2001 [11]. rFBA was tested on a simplified regulatory/metabolic network composed of 20 reactions and four regulatory proteins. This approach aimed to incorporate in the FBA analysis the regulatory effects that can impact the organism. Indeed, FBA can lead to incorrect predictions by not taking into account these regulatory effects; this was proved for *E. coli* in 2000 [46]. In rFBA, they added the regulatory effects in Boolean rules that define a new constraint on the

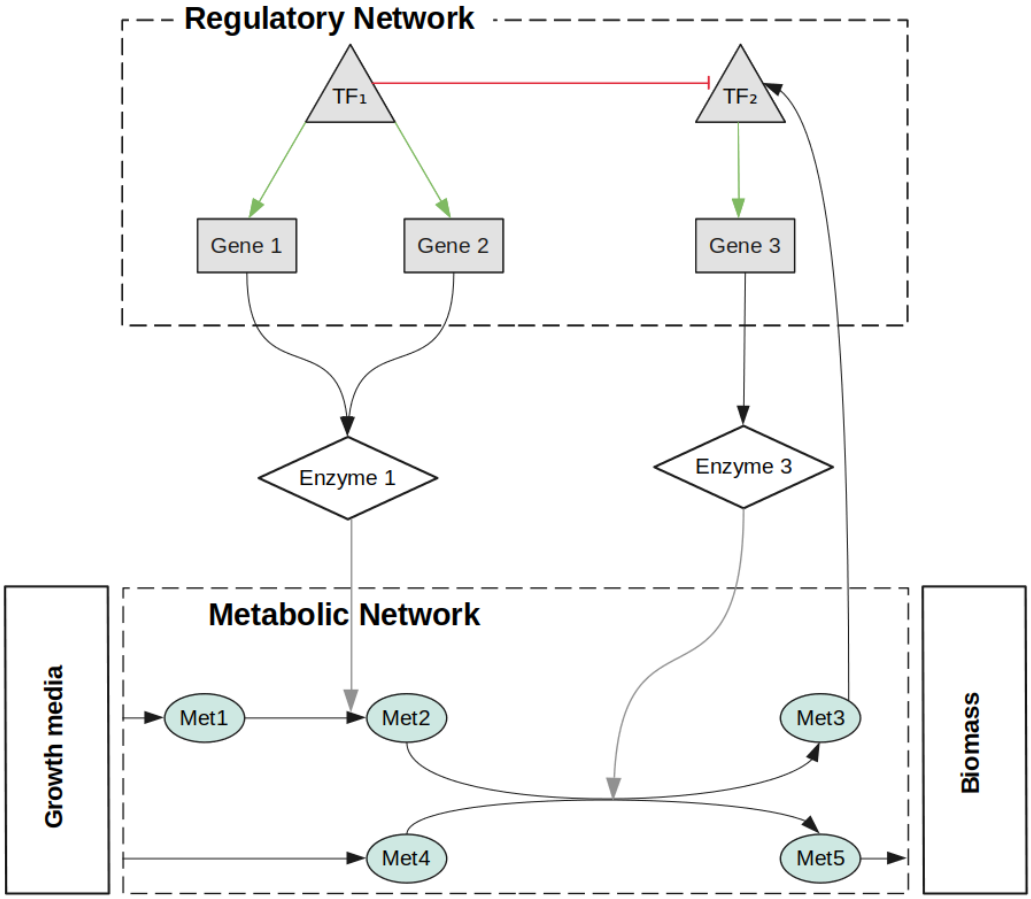


Figure 1.8 – **Link between regulatory and metabolic network** inspired by: [12]

system. rFBA updates the integrated metabolic-regulatory system for a short period by applying the defined Boolean rules for taking transcription and regulatory effects into account. These Boolean rules constrain reaction fluxes with enzyme presence/absence. The presence/absence of an enzyme is also defined by Boolean rules characterizing the presence/absence of given metabolites or activation/inactivation of a reaction. For example, if a reaction is catalyzed by an enzyme that is not present for a given time, then the flux of this reaction will be set to 0. At t=0, all initial system conditions must be given to do the update of the integrated metabolic-regulatory system. The updates are stopped when a given time is reached or there is a lack of an essential metabolite (e.g. carbon is exhausted). At this step, the optimal flux distribution for each reaction, the transport rates, and the extracellular concentrations are required. This approach allows a very detailed representation of a model closer to biological reality. The major downside is the need for many dynamic parameters that are not always known depending on the

organism. We also need a well-studied organism to construct the integrated network more accurately and infer the Boolean rules.

SR-FBA, steady-state regulatory flux balance analysis

This method is based on rFBA and was introduced by Shlomi et al. in 2007 [12]. It was used on a genome-scale integrated metabolic-regulatory model of *E. coli*. SR-FBA is based on Mixed Integer Linear Programming (MILP) used to identify a metabolic-regulatory steady-state (MRS). All the regulatory effects are expressed as Boolean rules; a gene or protein is given a Boolean value reflecting its expression. Then, these Boolean rules are translated into linear constraints added to the other constraints inside FBA. SR-FBA studies the effect of transcriptional regulation on cellular metabolism by quantifying the extent to which regulatory constraints and metabolic constraints determine the activity of fluxes. In addition, the integrated model is used to identify specific genes and metabolic functions in which regulation is not optimal. SR-FBA needs a well-known organism with an already integrated regulatory and metabolic network once again.

1.6.2 Probabilistic approaches

PROM, probabilistic integrative modelling

PROM was used in 2010 on *E. coli* and *M.tuberculosis* [13]. PROM uses gene expression data to infer conditional probabilities to represent gene states conditioned to the state of their regulators (or transcription factors, Tfs). PROM requires as input data a thousand gene expressions datasets (DNA chips, see Section 1.1) upon different conditions, a metabolic network, a list of the interactions between a TF and its targeted genes, and additional interactions involving enzyme regulation by metabolites and proteins. Using gene expression data and interaction data, we can model the effect of a perturbation on the regulatory and then on the metabolic network using PROM. However, PROM manages only direct interactions inside the regulatory network.

For example, predicting the impact of a knock-out (KO) of TF B over A, consists in computing $P(A = 1|B = 0)$, which will be estimated by counting the number of DNA chip samples in which target gene A is activated when TF B is deactivated. Similarly, the activation of TF B over A, $P(A = 1|B = 1)$, will be computed by counting the number of DNA chip samples in which target gene A is activated when TF B is activated. For modelling the perturbation of a TF on a genome-wide scale, the states of all target

genes are determined. The repercussion of a TF perturbation on the metabolism can be modeled passing by biochemical reactions. Indeed, the flow of biochemical reactions, which are regulated by genes interacting with the TF, will be constrained by the computed conditional probabilities of these genes. As an example, the maximum flux of the reaction regulated by gene A, which interacs with a knock-out TF B becomes:

$$fmax = p * Vmax$$

where p is the probability that gene A is turned ON when the regulatory TF B is turned OFF and $Vmax$ is the maximum rate of the reaction estimated by Flux Variability Analysis, FVA [47]. The new maximum flux $fmax$ is set as a soft constraint of the system. Indeed, the reaction can exceed $fmax$ with a penalty.

Probregnet, probability regulatory network

The Probregnet [5] pipeline [48] appeared in 2019 and was developed by Han Yu and Rachael Hageman Blair; the studied organism is *Human*. It is a complex global framework that allows integrating a gene regulatory model (based on graph interactions) into a metabolic network (based on biochemical reactions) using a constraint-based model. Probregnet gives as output for the metabolic network a new value of the objective function, see Section 1.5.1. This new value is impacted by an *in silico* perturbation done on the regulatory network.

The regulatory network analysis proposed by Probregnet is based on Bayesian networks, detailed in Section 1.4.2, also called probabilistic directed acyclic models [11]; which allows a representation of conditional dependencies between random variables. To illustrate Probregnet, the authors used as case-study the HIF-signaling pathway converted into a direct acyclic graph, DAG, where nodes are genes and edges are interactions between these genes (not signed or labeled). The BayesnetBP R package [49] is used to parametrize the graph with the gene expression data [50] by associating a node with its expression value. In a Bayesian network, the value of a child node depends on its parent nodes in the graph. Then, belief propagation is used to establish the repercussion of the perturbation of a given node in the graph over the other nodes. In [48] the perturbed node was HIF1A. The repercussion of the perturbation was monitored thanks to a ratio of the node expression in the perturbed model compared to the node expression in the

5. `https://github.com/hyu-ub/prob_reg_net`

model without perturbation. They focused on 15 enzymes present in the regulatory network, known to regulate biochemical reactions in the brain. The integration of the enzyme computational prediction to the metabolic network was done by using the same technique as in [51]. They performed a multiplication of the *in silico* enzyme predicted Fold-change with the flux of the biochemical reactions regulated by this enzyme. An average Fold-change was computed in the case of reactions regulated by multiple enzymes and this average Fold-change is then used for the multiplication. The fluxes of each reaction were obtained beforehand by performing an FBA; the objective function was the net ATP production. By doing this multiplication, the fluxes of the biochemical reactions can go out of the allowable solution space, as shown in Figure 1.7. They escape this problem by using mathematical paradigms such as LSEI (Least Squares with Equalities and Inequalities) or MCMC (Markov chain Monte Carlo) that allows them to take into account the new modified fluxes and stay in the allowable solution space by applying the two metabolic constraints defined in 1.5.1. In the end, they obtained a new net ATP production upon perturbation of HIF1A.

IDREAM, Integrated Deduced And Metabolism

IDREAM appeared in 2017 on *S. cerevisae* and was developed by Wang et al. [18]. IDREAM combines a statistical inference of regulatory influence network method (EGRIN) with PROM to create improved metabolic and regulatory network models.

A common strategy to link regulatory and metabolic networks is to use gene expression to impose condition-specific flux constraints on the model. The primary hypothesis is that elevated measurement of gene expression implies the increased activity of the metabolic enzyme encoded by that gene. The IDREAM authors observation is that in many cases, predictions using only FBA with maximum growth objective function are as good as methods using gene expressions to impose flux constraints on the metabolic network. They hypothesize that gene expression is not directly correlated with the encoded enzyme activity. Indeed, in some cases a gene expression and its encoded enzyme evolves differently under different conditions; for example, one increase or decrease and the other remain stable. This is the reason for using a regulatory network taking environmental influences or perturbations into account to relate them, as the one produced by EGRIN.

As mentioned before IDREAM uses EGRIN which stands for Environment and Gene Regulatory Influence [52]. EGRIN describes which factors influence gene expression and under which environmental conditions these factors intervene. EGRIN needs as input time

series transcriptome data under different environmental conditions and a prior network to have a partial knowledge of TF-target relationship.

The basic strategy of EGRIN is:

— **Identify co-regulated modules**. The subset of genes expressed coherently are identified as forming potentially co-regulated and coherent modules in certain environmental conditions. These modules can often be associated with specific aspects of cell function thanks to the gene ontology enrichment.

— **Identify directed TF-gene interactions**. First, a prior network is used to understand the existing interaction between genes and TFs. Then, these interactions are estimated with the expression dataset, and edges are removed from the prior network if the interaction does not happen in the dataset.

When compared to experimentally measured growth rates of *S. Cerevisiae* under different conditions, IDREAM is closer to these growth rates than PROM. However, it also needs thousands of expression data under different conditions to infer the regulatory network with EGRIN.

1.6.3 Synthesis

The techniques allowing an integration of the regulatory and metabolic network were developed recently, as the first approach was in 2001. Since this approach, many new approaches have appeared to do the integration. These approaches allow us to consider the regulatory effects that have a fundamental impact on metabolism. However, the majority of these approaches are used only on a well-known organism such as *E. coli* or *S. cerevisiae* because many input gene-expression profiles are required. Besides, some approaches do not scale well on a large-scale network, such as thousands of nodes network which is representative of the real size of a regulatory network, or need thousands of gene expression data to work. In Table 1.1, we show a summary where the approaches mentioned in Section 1.6 are compared in terms of the studied organism, principle, data required, and their advantages and inconveniences. The techniques presented here are the ones that come close to the method we will talk about later in this thesis.

Tools, organism	Input data	Principle	+ vs −
rFBA, 2001, *E.coli*	• Dynamic parameters • Integrated regulatory/metabolic network	• Translate regulatory effects into Boolean rules • Update the integrated system with these rules	+ : Considers multiple biological layers − : Not applicable on larger scale and unknown organisms
SR-FBA, 2007, *E.coli*	• Dynamic parameters • Integrated regulatory/metabolic network • Boolean rules for regulatory effects	• Regulatory effect translated into linear constraints • Add these constraints to FBA constraints	+ : Works on *E.coli* at a genome scale − : Applicable only on well-known organisms
PROM, 2010, *E.coli M.tuberculosis*	• Metabolic network • Regulatory network interactions • Thousands of gene expression datasets under different conditions	• Infer conditional probabilities to represent gene-TF interaction • Use this information to represent gene state • Add conditional probabilities as a constraint in FBA	+ : Does not need Boolean rules + : Applicable to a broader panel of organisms with uknown dynamic parameters − : Thousands of gene expression datasets
Probregnet, 2019, *Human*	• Regulatory network • Metabolic network • Gene expression dataset for at least 10 samples	• Use a Bayesian network to predict the effect of perturbation • Add these effects into FBA constraints • Apply mathematical paradigms (LSEI or MCMC) to estimate new ATP production	+ : Applicable on *Human* − : Does not handle inhibition interaction − : Requires several samples (at least 10) to ensure a statistical significance

Table 1.1 – **Summary of all the presented techniques which allow integration between regulatory and metabolic networks.**

1.7 Conclusion

This chapter introduces biological techniques used to generate gene expression datasets, biological networks and some approaches to model them and to model the integration between the regulatory and the metabolic network. These notions are important because they allow us to understand the complexity of modelling biological organisms and the repercussion of a perturbation on the different biological layers (e.g. regulatory, metabolic).

As said in Section 1.6.3, we face some challenges. More generally, there is a lack of

an integrated regulatory/metabolic approach that does not need as much input data as dynamic parameters or thousands of gene expression data, and can work on a large-scale network and a wider range of organisms and not only the most studied organism. For some organisms, dynamic parameters are not always available, or there is not thousands of gene expression data under different conditions.

In this thesis, we want to address these challenges with 2 contributions presented in Chapter 2 and Chapter 3.

Chapter 2 compares two gene regulatory modelling approaches; one uses a Bayesian network, Probregnet (see Section 1.6.2), and the other uses a logical network, Iggy (see Section 1.4.1). The logical and Bayesian approaches allows one to perturb the system and see the repercussion of this perturbation, and better comprehend the triggered biological mechanism. This comparison was made to enlighten the advantages and inconveniences of both techniques.

Chapter 3 addresses the logical approach's limitation regarding the new proposed node predictions upon perturbation and proposes a new approach called MajS. This new approach based on Iggy allows a more refined quantification of this prediction, which will be a crucial step in integrating the regulatory and metabolic network.

Both chapters focus on the regulatory network and propose computational predictions using only prior knowledge and observation data.

Chapter 2

First contribution : Comparison of Logic approach with Bayesian approach for Regulatory Network modelling

Summary of chapter 2

The impact of a given treatment over a disease can be modeled by measuring the action of genes on enzymes, and the effect of perturbing these last over the optimal biomass production of an associated metabolic network. In this chapter, we focus on presenting the comparison of two approaches: a logical (discrete) Iggy, and a probabilistic (quantitative) one Probregnet. Our objective was to compare the computational predictions of the enzymes in these models upon a perturbation. We used data from two previously published works that focused on the HIF-signaling pathway, known to regulate cellular processes in hypoxia and angiogenesis, and to play a role in neurodegenerative diseases, in particular on Alzheimer Disease (AD). The first study used Microarray gene expression datasets and the second one, used RNA-seq data. Our results on the Microarray dataset were that Iggy and Probregnet showed very similar (73.3% of agreement) computational enzymes predictions upon the same perturbation. On the second dataset, we obtained slightly different enzyme predictions (66.6% of agreement) using both modelling approaches; however, Iggy's predictions followed experimentally measured enzyme expression.

2.1 Introduction

This chapter is derived from our recently published study in the CMSB (Computational Methods in Systems Biology) conference in 2020 [14]. The aim of this study is to compare a recent tool Probregnet [48] with Iggy [8]. Probregnet uses a Bayesian network model, on which belief propagation techniques are applied to reason over it. Iggy uses a sign-consistency approach, expressed as a logic program in Answer Set Programming [44]. Both tools use prior regulatory knowledge and are able to make computational predictions upon system perturbation using few gene expression datasets. The nature of both approaches, one quantitative, the other discrete, makes it interesting for us to compare them in the context of enzyme prediction.

The results of this comparison were obtained on the HIF signaling pathway, known to be of major importance in neurodegenerative diseases [50]. We applied both tools on a Microarray dataset on Alzheimer's disease and an RNA-Seq dataset on human umbilical vein endothelial cells. We built models upon two regulatory networks of around 80 nodes and 250 edges.

Iggy is faster than Probregnet to compute enzyme predictions in our tested case studies (0.038s vs 25s) [1]. Besides, Iggy and Probregnet showed very similar (73.3% of agreement) computational enzymes predictions upon the same perturbation for Microarray data. On the second dataset, we obtained different enzyme predictions (66.6% of agreement) using both modelling approaches; however Iggy's predictions followed experimentally measured results on enzyme expression.

2.2 Methods

2.2.1 Datasets to perform this comparison

We used two different datasets. The first one is a Microarray gene expression dataset published in [50] and the second one is an RNA-seq dataset published in [53]. The DNA microarray sequencing and RNA-Seq general techniques are explained in Section 1.1 of Chapter 1.

The Microarray data were measured in the Hippocampus brain region of 10 Alzeihmers's patients and 13 healthy patients. The Hippocampus is known to be differentially

1. All computations were performed on a standard laptop machine. Ubuntu 18.04, 64 bits, intel core i7-9850H CPU 2.60 GHz, 32 GB

vulnerable to the histopathological and metabolic features of Alzheimer's disease (AD). An Affymetrix Human Genome Array was used and allowed to collect the expression for 20545 genes.

The RNA-seq data were measured on human umbilical vein endothelial cells (HUVECs) exposed to constitutively active HIF1A over-expression. This data was collected for 3 control cells (with normal expression of HIF1A) and 3 cells with induced over-expression of HIF1A in the form of two types of RNA-seq datasets, one absolute and the other differential. The absolute RNA-seq datasets, consisting of 25691 RNA, were normalized using the edgeR R package [54]. These normalized RNA-seq data were used to generate the *in silico* predictions with Probregnet. The differential RNA-seq datasets were composed of 1854 genes significantly differentially expressed upon HIF1A induction. The genes having a significant differential expression were selected using a cutoff of 1.5, applied on their logarithmic expression. A cutoff of 0.01 was used on the false discovery rate (FDR). This differential RNA-seq dataset was used to generate the *in silico* predictions with Iggy. All the RNA-Seq datasets were extracted from the GEO database [2].

2.2.2 Regulatory network

In [55] it has been shown that the HIF-signaling pathway is of major importance in neurodegenerative disease, with a key role of the HIF1A protein. In [48] the authors built a gene regulatory network for Alzheimer Disease (AD), focused on the HIF-signaling pathway. We used a signaling and gene regulatory network built upon the same pathway; for this purpose we use the same methods as proposed in the Probregnet pipeline. These steps are explained in the following paragraphs. The retrieved networks were afterward modeled and analyzed with Probregnet and Iggy using two different datasets (see Section 2.2.1).

At first, the HIF-pathway was extracted from the KEGG database thanks to the *graphite* R package [3]. This R package allows to provide networks derived from different databases based on the pathway topology. In this network, all the metabolites nodes have been removed and the edges are propagated through them, and are labeled as indirect processes. The nodes represent either protein or genes. The edges represent multiple biological processes: ubiquitination, phosphorylation, binding, inhibition, activation, expression.

2. `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE98060`
3. `https://www.bioconductor.org/packages/release/bioc/vignettes/graphite/inst/doc/graphite.pdf`

Afterward, we reduced this graph by keeping only the nodes of the network associated with genes present in the gene expression datasets. Since we had two datasets we will retrieve in these step two reduced networks. The first one was based on the Microarray data in [50] extracted from the Hippocampus brain region of healthy and Alzheimer's disease (AD) patients. The genes which were kept were those present in either healthy or AD datasets. In the second network, obtained using the RNA-seq dataset of HUVECS [53], the genes kept were those that were present in either control or HIF1A induced cells.

Finally, both regulatory networks were converted into a directed acyclic graph (DAG) by using the *pcalg* R package [4] that allows to extend a partially directed acyclic graph into a DAG using the algorithm by Dor and Tarsi (1992) [56]. In this algorithm, the DAG will have the same set of vee-structures as the partially directed acyclic graph; where a vee-structure is formed by two edges, directed towards a common head, while their tails are non-adjacent. After this process, the edges in the DAG are not labeled (or signed) anymore. Since Iggy, contrary to Probregnet, needs a signed graph, we took into account the edges that were previously labeled as inhibition in the KEGG database, and the other edges were all labeled as activation. The final regulatory network consists of 94 nodes and 285 edges for the Microarray data and 81 nodes and 233 edges for the RNA-seq data. Both are a reduction of the HIF-signaling pathway adapted to the data.

2.2.3 The Probregnet pipeline

The Probregnet pipeline [48] allows to integrate a gene regulatory model, using Bayesian network (see Section 1.4.2), into a metabolic network using a constraint-based model, FBA (see Section 1.5.1). This pipeline is detailed in Section 1.6.2. The regulatory network used as case study represents the HIF-1 signaling pathway known to have an impact on neurodegenerative disease and the metabolic network is composed of 71 biochemical reactions, known to be active in brain metabolism. They focus on the prediction of 15 enzymes upon perturbation of HIF1A. These 15 enzymes are catalyzing 10 reactions among the 71. In the results presented in this chapter, we still focus on these 15 enzymes and compute the ratio (or fold-change) of the enzymes' expression in a perturbed model compared to the enzymes' expression in a model without perturbation. For this, we used different Bayesians networks (BNs) parametrized for the two different datasets presented in Section 2.2.1.

4. `https://cran.r-project.org/web/packages/pcalg/pcalg.pdf`

Bayesian networks for our case-studies

Recall that the BN is built using the DAG extracted in Section 2.2.2 parametrized to a specific dataset. Using these DAGs, we obtained from our two datasets (AD and HUVECs) the following BNs:

1. Microarray dataset of the Hippocampus brain region
   (a) BN parametrized using the Microarray data of the 10 AD patients
   (b) Control BN parametrized using the Microarray data of the 13 healthy individuals
2. RNA-seq dataset of HUVECs
   (a) BN parametrized using the RNA-seq data of the 3 adenovirally over-expressed HIF1A cells
   (b) BN parametrized using the RNA-seq data of the 3 HUVECs with normal HIF1A expression

For BN (2a) and BN (2b), the number of cells was not enough for Probregnet in order to parametrize the BN. Therefore, for each of the two conditions (normal and adenovirally over-expressed HIF1A), we completed the 3 HUVECs datasets with 10 artificially generated datasets (by adding an artificial noise in the data of 1%).

Enzymes *in silico* predictions

For the Microarray dataset of the Hippocampus brain region we computed the fold-change of the 15 enzymes for different types of *in silico* perturbations of the model. Equation 2.1 describes the expression ratio measured for each enzyme e.

$$y_e = \frac{x_e^{AD_p}}{x_e^C} \tag{2.1}$$

where y_e refers to the fold-change (FC) expression of enzyme e; $x_e^{AD_p}$, to the expression of enzyme e obtained after simulation of the AD BN (1a) upon perturbation p. This perturbation p was done in three ways: HIF1A over-expressed (set to 13 expression level), under-expressed (set to 8 expression level), and HIF1A unaltered (9.56 expression level). For HIF1A unaltered, enzyme expression is the average expression of the enzyme in the dataset for AD patients. x_e^C refers to the expression of enzyme e in the control BN (1b) without perturbation, that is, the average expression of the enzyme in the dataset for

healthy patients.

For the RNA-Seq dataset on HUVECs we still focus on the enzyme and only one *in silico* perturbation. Equation 2.2 describes the fold-change computed for each enzyme e.

$$z_e = \frac{x_e^{O_p}}{x_e^H} \tag{2.2}$$

where z_e refers to the FC expression of enzyme e, $x_e^{O_p}$ corresponds to the expression of enzyme e obtained after simulation of the HUVECs over-expressed BN (2a) upon perturbation p. This perturbation p represents an over-expression of HIF1A (set to 17 expression level, when HIF1A average expression across over-expressed samples is 14.5). x_e^H refers to the expression of enzyme e in the BN (2b) without perturbation, that is, the average expression of the enzyme in unaltered cells.

2.2.4 Iggy

The Iggy framework allows to model a regulatory model, using Answer set Programming based on a sign-consistency approach. Iggy automatically detects inconsistencies between a graph G and a set of experimental observations μ, applies minimal repairs to restore consistency, and predicts the sign of non-observed nodes in G. This framework is more detailed in Section 1.4.1.

Generating discrete observations from datasets to use Iggy

For the Microarray dataset, we denote as $\bar{y}_g$ the ratio (or fold-change) of the average expression of gene g of AD patients against the average expression of gene g in healthy individuals. To obtain the associated sign for each gene g in the dataset, we discretised $\bar{y}_g$, using the thresholds over the distribution of the expression of the 20545 genes in the dataset as shown in Equation 2.3.

$$sign(\bar{y}_g) = \begin{cases} + & \text{if } \bar{y}_g > Q_3 \\ - & \text{if } \bar{y}_g < Q_1 \\ 0 & \text{if } 0.99 \leq \bar{y}_g \leq 1.01 \end{cases} \tag{2.3}$$

where Q_3 and Q_1 refer to the third and first quartiles of the fold-change gene expression data distribution. From this discretisation analysis, the input observations data for Iggy was composed of 16 "+", 24 "-", and 16 "0-changed" nodes. Nodes not included in these

thresholds are seen as not significantly observed and therefore do not belong to the set of observed nodes. This set, denoted as μ_1, did not include any of the 15 enzymes that will be computationally predicted. Besides, the sign of 3 other nodes (EP300, CREBBP, ARNT in Figure 2.2), which are direct predecessors of the enzymes, is set to "0-change" in μ_1 so that we can see only the impact of HIF1A on the enzymes.

To simulate a perturbation in Iggy, there is no change in the regulatory network structure, however the set of observations μ_1 changed slightly:

$$\begin{aligned} S^+ &= \mu_1 + (sign(\bar{y}_H) = +) \\ S^- &= \mu_1 + (sign(\bar{y}_H) = -) \\ S^0 &= \mu_1 + (sign(\bar{y}_H) = 0) \end{aligned} \tag{2.4}$$

where $\bar{y}_H$ refers to the expression level of HIF1A. We built then three sets of observations, denoted as S^p, where p refers to the type of sign imposed to the HIF1A node to simulate an over-, or under-expression of HIF1A, as well as a *non-change* effect of this protein.

For RNA-Seq data, we used the logFC between HIF1A over-expressed and normally expressed already present in the gene differentially expressed data from RNA-Seq analysis (see Section 2.2.1). We denote this logFC for each gene g as $\bar{z}_g$. To transform the quantitative value of $\bar{z}_g$ in signs we used the same logic as before but with thresholds better adapted to this dataset (Equation 2.5).

$$sign(\bar{z}_g) = \begin{cases} + & \text{if } \bar{z}_g > 1.5 \\ - & \text{if } \bar{z}_g < -1.5 \\ 0 & \text{if } -0.15 \leq \bar{z}_g \leq 0.15 \end{cases} \tag{2.5}$$

From this new discretisation analysis, the input observations data for Iggy was composed of 5 "+", 2 "-", and 19 "0-changed" nodes. Nodes not included in these thresholds are seen as not significantly observed and therefore do not belong to the set of observed nodes. This set, denoted as μ_2, did not include any of the 12 enzymes that will be computationally predicted and the 3 nodes (EP300, CREBBP, ARNT in Figure 2.2) that are direct predecessors of the enzymes are still set to "0-change". Only 12 enzymes were kept and not the 15 initially as three of them (HK3, ENO3 and PDHA2) were not considered to be expressed in the study. Indeed, using RNA-Seq techniques, we obtained reads for each transcript and can quantify an RNA activity using the number of reads which represent this specific RNA (see Figure 1.2). Some RNAs are seen as unexpressed if the

number of reads is under a threshold that is fixed to 10 in count-per-million, scaled by the total number of reads. HK3, ENO3, and PDHA2 are under this threshold, so we removed them from the gene regulatory network [57]. From μ_2 we built two sets of observations, R^+ and R^0, where the sign imposed to HIF1A, $\bar{z}_H$, was either "+" (over-expressed) or 0 (unaltered), as described by Equation 2.6.

$$\begin{aligned} R^+ &= \mu_2 + (sign(\bar{z}_H) = +) \\ R^0 &= \mu_2 + (sign(\bar{z}_H) = 0) \end{aligned} \tag{2.6}$$

where $\bar{z}_H$ refers to the expression level of HIF1A. As for the case of Probregnet, we focused on the Iggy's *in silico* prediction of the 12 enzymes upon HIF1A perturbations in the system. We recall in Figure 2.1 the different steps described in this Section for Iggy and Probregnet. All scripts and data described in this chapter are available at: `https://gitlab.univ-nantes.fr/E19D080G/comparing_iggy_prob.git`

2.3 Results

We focused on the *in silico* computational predictions from both approaches on enzymes involved in biochemical reactions of brain metabolism upon HIF1A stimulation. We illustrate our results in two case studies. The first, uses Microarray gene expression data from the Hippocampus brain region of Alzheimer's Disease (AD) patients and healthy individuals. The second, uses RNA-Seq data of 6 Human umbilical vein endothelial cells (HUVECs) over-expressing adenovirally HIF1A protein or expressing normally HIF1A.

2.3.1 HIF1A impact on HIF-signaling pathway for Alzheimer's Disease patients

The Microarray data used for this case-study is presented in Section 2.2.1. The network, corresponds to the HIF signaling pathway (see Section 2.2.2). Both data, gene expression datasets and network, were transformed (see Sections 2.2.2, 2.2.3 and 2.2.4) in order to be used for the comparison of Iggy and Probregnet.

HIF signaling pathway

We chose the HIF signaling pathway and focused on the HIF1A protein, which is a potential therapeutic target for neurodegenerative disease [55]. The HIF network, obtained

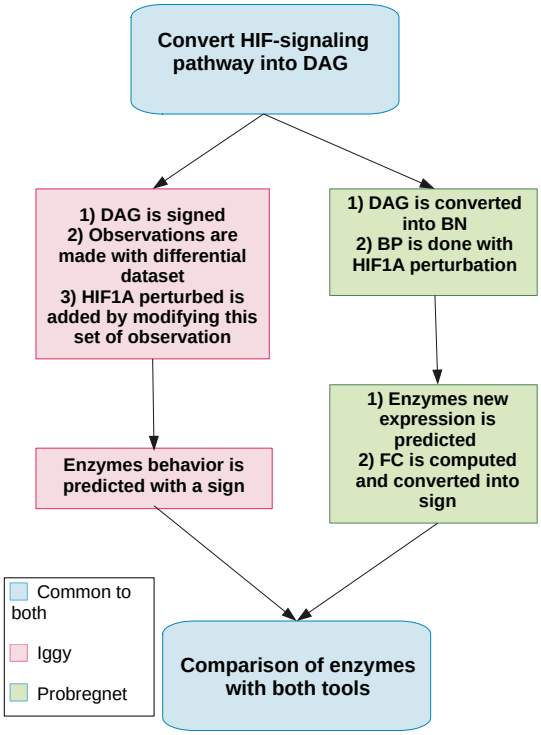


Figure 2.1 – **Diagram representing the different steps in order to compare the two approaches.** The steps for both are in blue, those specific to Iggy are on the left in red and those specific to Probregnet are on the right in green. DAG stands for directed acyclic graph, BN for Bayesian network and BP for belief propagation.

in [48], was extracted from the KEGG database and then reduced (see Section 2.2.2). The resulting graph from this network (94 nodes, 285 edges) was built from the experimental data. The nodes represented genes and proteins, while the edges represented signaling and gene regulatory interactions. In Figure 2.2 we show a subgraph of this HIF graph, focusing on the genes of the network that are directly connected to the enzymes.

Evolution of enzyme production according to HIF1A fluctuation with the Bayesian approach

We present here the results obtained with the Probregnet pipeline (see Section 2.2.3 and Table 2.1). We compared three particular (perturbed) states with respect to an *unaltered state* of the system, and computed the predictions of the fold-change of the enzymes

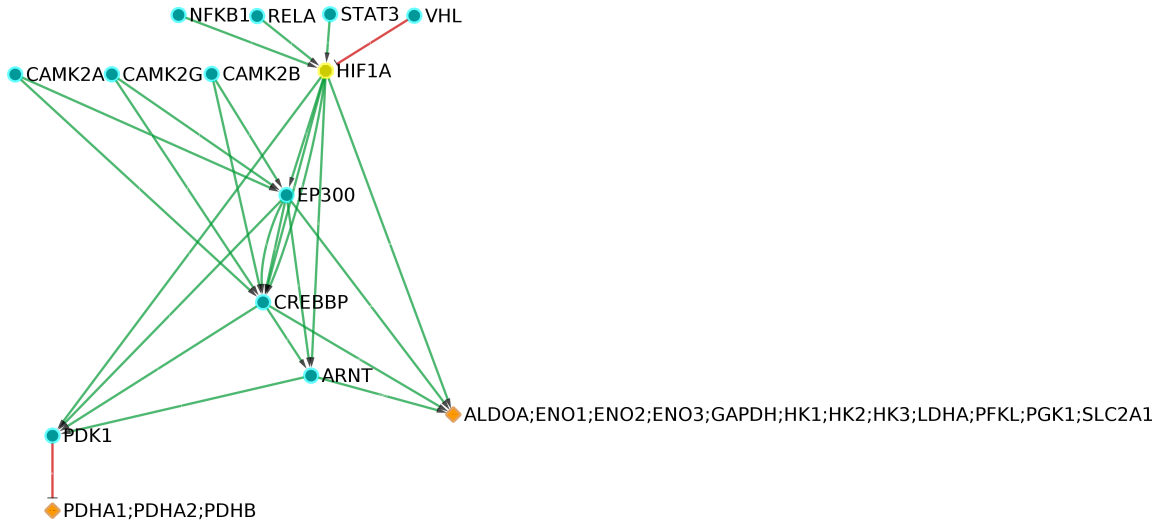


Figure 2.2 – **Subgraph regulatory network of HIF-pathway**. Only the enzymes and their predecessors are represented in this schema. The enzymes are represented as orange diamonds, the predecessors genes as blue circles, and the perturbed node, HIF1A, as a yellow circle. The edges represent either activation in green or inhibition in red.

level for each comparison (see Equation 2.1).

Name	Description
HIF1A -	AD model with HIF1A under-expressed (HIF1A expression set to 8) against healthy model without perturbation (HIF1A normal expression)
HIF1A 0	AD model without perturbation (HIF1A normal expression) against healthy model without perturbation (HIF1A normal expression)
HIF1A +	AD model with HIF1A over-expressed (HIF1A expression set to 13) against healthy model without perturbation (HIF1A normal expression)

Table 2.1 – The three compared model states. The name of this comparison, used in the rest of this Section, appears in the first column.

As we can see in Figure 2.3, the predicted fold-change of 9 out of 15 enzymes increases across the three comparative states of the system ordered as:
HIF1A - , *HIF1A 0*, *HIF1A +*.

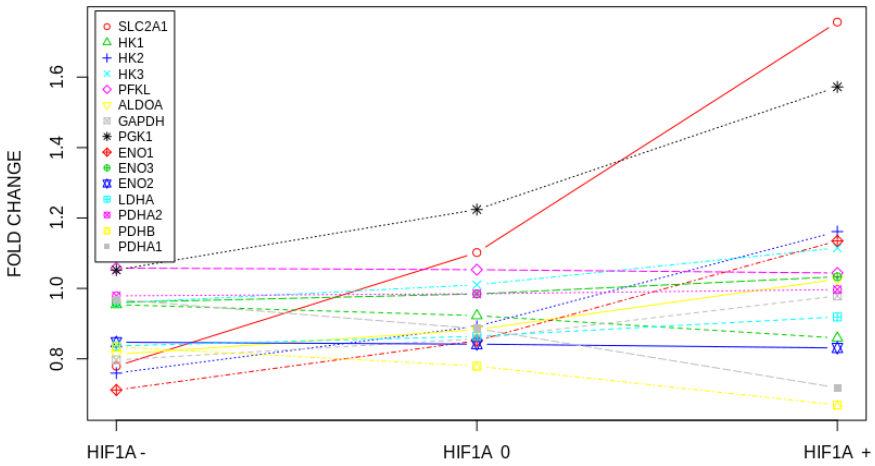


Figure 2.3 – **Probregnet computational predictions using three perturbed states of the HIF model.** Evolution of the fold-change of the 15 enzymes across the perturbed system states detailed in Table 2.1. In the X-axis we show the 3 perturbed states of the system, in the Y-axis, the value of the predicted fold-change.

Evolution of enzyme production according to HIF1A fluctuation with the logical approach

Here we used Iggy with the same regulatory network as Probregnet with 3 sets of observations (see Equation 2.4) that correspond to the genes variations in each of the three perturbed states (see Table 2.1). Our results (see Table 2.2), focus on the *sign prediction* of the 15 enzymes. The sign represents the over-expression ("+", green), under-expression ("-", red), and the no-variation (0, blue) of the level of the enzymes upon each comparative case detailed in Table 2.1. All but three of the enzymes are over-expressed when HIF1A is over-expressed. The three enzymes that are evolving with a contradictory sign are the ones inhibited by PDK1 (see Figure 2.2), this goes in agreement with the sub-graph topology.

Comparison of the enzymes computational predictions using Iggy and Probregnet

Recall that Iggy predicted discrete signs of the nodes in the graph whereas Probregnet, quantitative values. Thus, for each enzyme, we compared Iggy's predicted sign against the

HIF1A -	**HIF1A** 0	**HIF1A** +
PDHA1 = +	PDHA1 = 0	PDHA1 = -
PDHA2 = +	PDHA2 = 0	PDHA2 = -
PDHB = +	PDHB = 0	PDHB = -
LDHA = -	LDHA = 0	LDHA = +
GAPDH = -	GAPDH = 0	GAPDH = +
HK1 = -	HK1 = 0	HK1 = +
HK2 = -	HK2 = 0	HK2 = +
HK3 = -	HK3 = 0	HK3 = +
ENO1 = -	ENO1 = 0	ENO1 = +
ENO2 = -	ENO2 = 0	ENO2 = +
ENO3 = -	ENO3 = 0	ENO3 = +
PGK1 = -	PGK1 = 0	PGK1 = +
SLC2A1 = -	SLC2A1 = 0	SLC2A1 = +
PFKL = -	PFKL = 0	PFKL = +
ALDOA = -	ALDOA = 0	ALDOA = +

Table 2.2 – **Iggy's sign prediction of the 15 enzymes after perturbing HIF1A.**

derivative sign of the mathematical curve represented in the plots of Figure 2.4. If the sign of the derivative is the same as the tendencies observed for Iggy in the 3 comparisons, then the name of the enzyme will appear in green, else, in red. 11 enzymes will evolve in the same way with the two approaches except for HK1, PFKL, ENO2 and PDHA2. Probregnet fold-change expression of 3 out of 4 of these enzymes will remain unaltered (difference in fold-change expression of less than 0.1) across the three comparative cases. Besides, the probabilistic approach does not take the inhibiting effect of PDK1 on the three PDH enzymes into account as it adds a manual correction by multiplying the fold-change of these three enzymes by the inverse of the fold-change predicted for PDK1 in [48]. The only one that is significantly decreasing and opposite to Iggy's prediction is HK1.

2.3.2 *In vitro* over-expression of HIF1A in HUVECS (human umbilical vein endothelial cells)

The induced over-expression of HIF1A adenovirally allows us to do a comparison between Iggy and Probregnet with another dataset, for which experimental perturbation results are available.

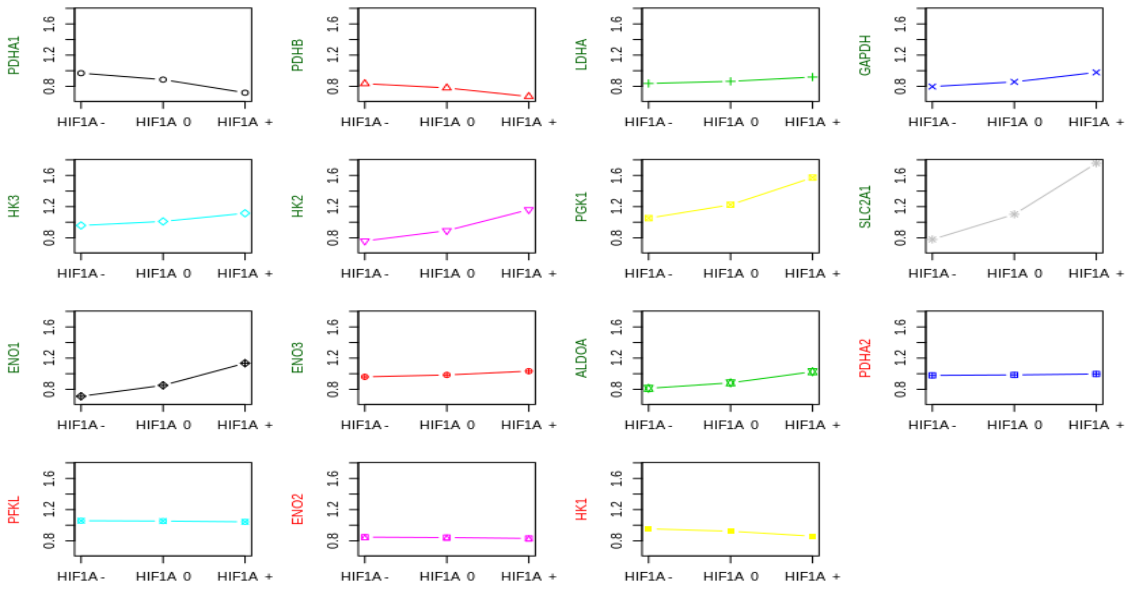


Figure 2.4 – **Probregnet fold-change evolution (Y-axis) for each of the 15 enzymes compared to Iggy's predicted sign**. Three comparative cases are studied corresponding to *HIFA1 -*, *HIF1A 0*, and *HIF1A +*, as detailed in Table 2.1 (X-axis). The 11 enzymes in green are evolving in the same way as the predicted sign of Iggy, while the 4 red ones are evolving in a different way.

Regulatory network from HIF signaling pathway

As explained in Section 2.2.2, we converted the HIF signaling pathway into a regulatory network adapted to the RNA-Seq data. We obtained a new regulatory network of 81 nodes and 233 edges. Its structure is strongly similar to the precedent one and the enzymes neighbourhood is the same as Figure 2.2. The main difference is that there are new regulators of HIF1A in this regulatory network (7 nodes are predecessors of HIF1A and not only 4 as shown in Figure 2.2).

Comparison between real experimental data and Iggy's and Probregnet computational predictions

We used the absolute normalized RNA-Seq dataset for Probregnet; while the differentially one for Iggy (see Section 2.2.1). The studied condition was the comparison between the enzymes expression in a model with HIF1A protein induction with respect to a model without HIF1A induction. Once the graph was made and data transformed we were able to apply Probregnet and Iggy on these data. Our results are shown in Figure 2.5. We

exclude for this study the enzymes HK3, ENO3 and PDHA2 because their expression level was too low in HUVECs cells. Therefore, we will study the expression of only 12 enzymes.

For Probregnet predictions (blue bars in Figure 2.5), we used the normalized dataset (Section 2.2.1) and computed the fold-change for each enzyme (see Equation 2.2). For Iggy predictions, we generated a new set of observations (see Equation 2.6) and computed the predictions for each enzyme. Recall that the 12 enzymes sign was not contained in the observation dataset. In Figure 2.5, we present only the Iggy predictions using the observation dataset R^+ (see Section 2.2.4).

We obtained 10 "+" predictions and 2 "-" predictions in the PDHA1 and PDHB enzymes. The observation dataset R^0, generates "0" predictions (unchanged behaviors) for all of the 12 enzymes.

For the experimental observations (pink bars in Figure 2.5), we used the normalized dataset and computed the fold-change of each enzyme as the average enzyme expression across HIF1A induced cells against the average enzyme expression across normal cells. In Figure 2.5 we can see that the enzyme levels evolve in the same way for Iggy and the real experimental data but slightly differently with Probregnet (8 of 12 have the same tendency). In addition, we compared all the signs predictions for all the nodes present in the graph (81) and Iggy predicted 65% in the same way as the real data, while Probregnet only 43.75%.

2.3.3 Quantification of Iggy's predictions

Our final aim is to provide an integration of the computational predictions to constrain the metabolic fluxes. The integration between the regulatory and metabolic network via Probregnet was done using the quantitative values of the computational predictions. The integration is done in several steps with Probregnet. The first step is by performing multiplication between the *in silico* perturbed enzymes expression and the reactions catalysed by these enzymes (see Section1.6.2). The Iggy predictions being qualitative, it was necessary to define an appropriate integration scheme. In order to integrate Iggy into the metabolic network, as done in Probregnet, our first challenge was to quantify the qualitative predictions of Iggy. We designed two methods: the labellings and the thresholds methods, which are detailed in the following paragraphs to tackle this challenge.

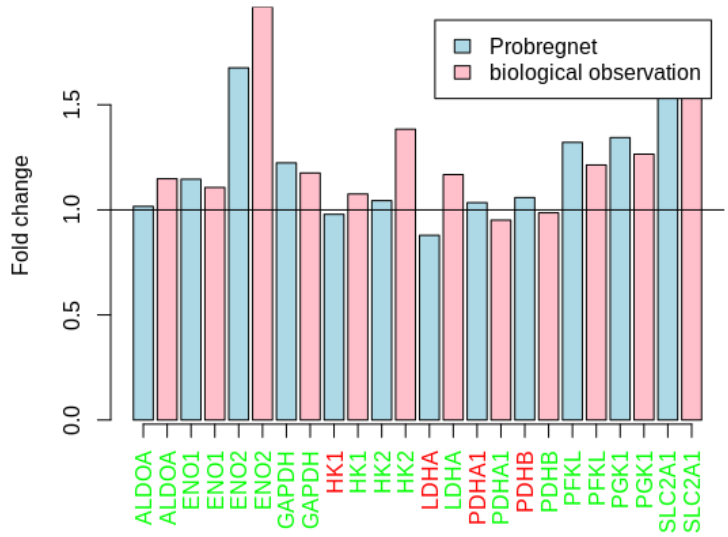


Figure 2.5 – **Comparison between Probregnet, biological observation and Iggy.** The FC of each enzyme computed with Probregnet is represented by blue bars, while the FC of biological observation by pink bars. If the Probregnet prediction or the biological observation of the enzyme agrees with Iggy's prediction sign, that is $FC > 1$ agrees with "+" and $FC < 1$ agrees with "-", the enzyme name is colored in green, else, in red.

The labellings method: deducting a percentage associated with a sign by analysing the labellings

The first experiment conducted is the implementation of new parameters in Iggy. The implementation of this functionality on Iggy is available on:`https://github.com/bioasp/iggy/tree/issue11`. This new implemented parameter is noted -c (–count_labelings), which allows knowing the total number of labellings (solution space) without enumerating all of them. This enumeration was the default parameter, -l (–show_labelings). Using these two parameters (-l and -c), we deduced the percentage associated with nodes' signed predictions with the two following steps:

— First count for a node the number of labellings related to a sign. This number is obtained by analyzing the solution space (-l parameter).

— Then, this number is divided by the total number of labellings (-c parameter).

In the end, we obtained a sign associated with a percentage for each node. This method is relevant for weak predictions (see Section 1.4.1) as it can give us an idea of the most probable sign assigned to a node by considering the associated percentage. This method is also helpful on non-predicted nodes as it can give us an idea of the most probable sign regarding the associated percentage. However, for strong predictions, this method is not adding any information as the associated percentage with the predicted sign will always be 100%. An example of this method results is illustrated in Figure 2.6 on the AD case study graph composed of 94 nodes and with HIF1A set to "0".

By exploring the solution space composed of 270 labellings, we observe that six nodes vary through the labellings (those whose lines oscillate). Among these six nodes, five are nodes not predicted by Iggy and one node, STAT3, is predicted as a "CHANGE" by Iggy (see Section 1.4.1). We can deduce a percentage for the varying nodes by analysing these labellings. We, therefore, have two unpredicted nodes, ERBB2 and INSR, which have a percentage across the labellings of 60% "-", 20% "+", and 20% "0". The three other unpredicted nodes, PIK3CD, PIK3R2 and RPS6, have a percentage of 33% "-", 33% "+", and 33% "0". The node predicted as CHANGE by Iggy varies across the labellings of 50% "+" and 50 % "-". For this case study, by analysing the total number of labellings, we can add information for two unpredicted nodes (among 5), which are most likely to be signed as "-".

In the end, this method allows us to add information only on some nodes which are either weak predictions or non-predicted and where we have an associated percentage with a majority. Moreover, analysing all the answer sets is not always possible. In most of the studied cases, we cannot explore all the solution space that is way too big. For example, the AD case study when HIF1A = - (see Table 2.1) outputs a set of 10^{10} labellings after 21 days of computation without giving a complete solution. Therefore, this idea seemed unlikely to generate quantitative predictions for most case studies. Besides, as illustrated in 2.6, having only the first labellings is not enough to deduct a percentage. For this case, if we had only the ten first labellings, some nodes such as INSR will be set to "+" with the percentage of 100% which is not the case when analysing all labellings.

The thresholds method: using thresholds based on the observation to discretise Iggy's prediction

Our second attempt to quantify Iggy's results was performed by using different thresholds to discretise Iggy's prediction. The thresholds are the same as the ones taken to

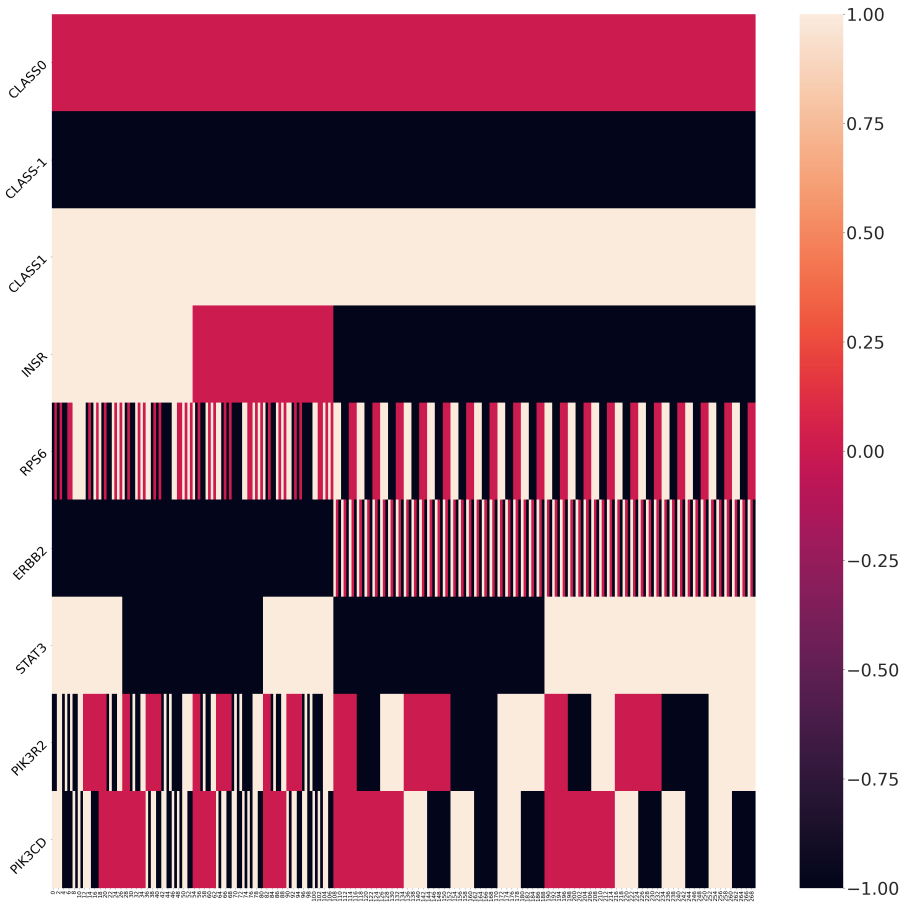


Figure 2.6 – **Heat map representing the total number of labellings for AD case study with HIF1A 0**. The x-axis represents the labellings from 1 to 270 for this case study. The y-axis shows the nodes of the network which are varying across the labellings (6 in total), the other nodes (among the 94) are divided into 3 different classes which do not vary. CLASS0 for the nodes which are always at "0" (13 nodes), CLASS1 for the nodes which are always at "+" (47 nodes) and CLASS-1 for the nodes which are always at "-" (28 nodes). In black, the sign is set to "-", in red to "0", and in beige to "+".

generate the discrete observation for the AD datasets (see Section 2.2.4). Recall that the thresholds are deducted from the distribution of the expression of the 20545 genes. All nodes predicted to "+" are set to Q_3 (1.0407). The ones predicted to "-" are set to Q_1

(0.9542), and the ones predicted to "0" are set to 1. These new quantified predictions of Iggy are denoted as $FC_{(Iggy)}$. Q_3 and Q_1 refer to the third and first quartiles of the fold-change gene expression data distribution.

2.3.4 Integration of the regulatory and metabolic networks

In this section, we present how the enzymes' computational predictions can be used to constrain the metabolic network. First, we provide information about the Probregnet pipeline's integration process. Then, we adapt this process to consider Iggy's qualitative predictions. Finally, we compare the results obtained for both methods.

The Probregnet pipeline integration with the metabolic network

The probregnet pipeline integration with the metabolic network is illustrated in Figure 2.7. In [48], they worked with the RECON1 [58] human metabolic network model and focused only on 71 reactions which have an impact on brain metabolism. The authors perform an FBA to study the net ATP production. After the FBA computation, an optimal flux for each of the 71 reactions was obtained. These fluxes are related to the behavior of a healthy brain. In order to integrate the metabolic model with the regulatory network, they focused on 15 enzymes, present in the regulatory network, known to regulate the reactions inside this metabolic network. The repercussion on the enzymes upon HIF1A perturbation, is monitored thanks to a ratio called Fold-Change (FC), which represents the node expression in the perturbed model divided by the node expression in the model without perturbation. Then, the integration is done by multiplying the *in silico* enzymes predicted FC with the flux of the reactions catalysed by these enzymes. This method was proposed in [51]. Precisely, given the flux of a reaction i in a healthy brain (denoted f_i) and the mean Fold-Change of the enzymes regulating reaction i (denoted $\overline{FC}$); we define the new flux of reaction i (f_i') by the formula:

$$f_i' = f_i * \overline{FC} \tag{2.7}$$

The Fold-Change of the enzymes is obtained with the Probregnet regulatory network analysis.

In this way, by focusing on the 15 enzymes, 10 out of 71 reactions are modified as presented in Equation 2.7. When conducting this multiplication, the fluxes of the biochemical reactions can go out of the allowable solution space (see Section 1.5.1). In the Probregnet

pipeline, they applied two techniques to prevent this problem. The first is a linear program optimisation approach, namely LSEI (least squares with equalities and inequalities) and the second a simulation MCMC (Markov chain Monte Carlo). More generally, these mathematical paradigms aim for each i fluxes of the ten modified reactions to be as near as possible to the value given by Equation 2.7 (f'_i) but subject to the typical constraints in FBA (mass balance and capacity constraints).

After applying one of the mathematical paradigms, either LSEI or MCMC, they obtained a new value of net ATP production considering the perturbation.

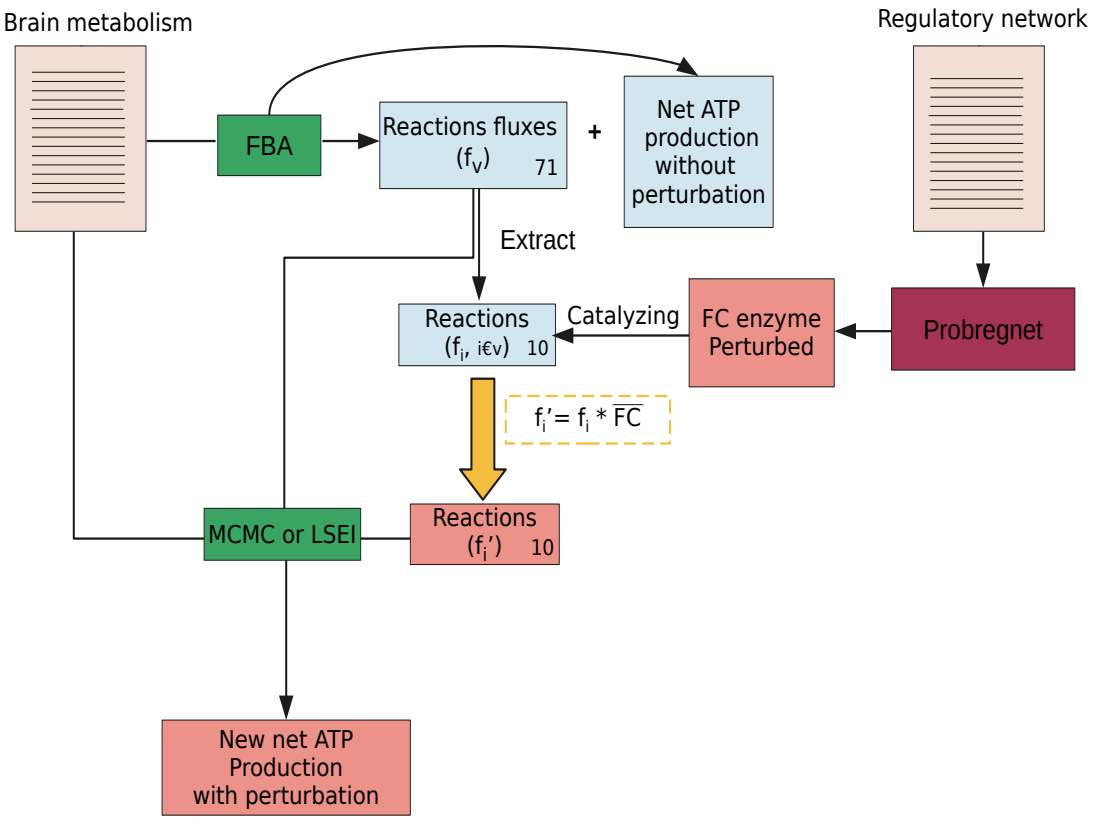


Figure 2.7 – **Probregnet pipeline integration with the metabolic network**. In light blue, the model without perturbation. In light red, the perturbed model. In orange, the mathematical transformation applied. In green, mathematical paradigms applied. In purple, method for modelling regulatory networks applied. In beige, the input data. The number at the bottom-right of the "Reactions" box means the number of reactions inside.

Impact of Iggy prediction with HIF1A perturbation over net ATP production compared to Probregnet prediction

In order to integrate Iggy's prediction with Probregnet metabolic analysis, we focused on the quantified prediction of Iggy, $FC_{(Iggy)}$ (see the thresholds method, Section 2.3.3). We multiplied the average Iggy's quantified predictions of enzymes, $\overline{FC}_{(Iggy)}$, with the flux of the reactions catalysed by these enzymes, and we computed the new fluxes (see Equation 2.7).

Following this step with Iggy's quantified predictions, we obtain the same number of modified reactions (10 out of 71) but with different modified fluxes (f_i''). Then we also apply an LSEI or MCMC to stay in the allowable solution space (see Section 1.5.1).

Therefore, we obtain a new value for the objective function, namely the net production of ATP, which considers the different perturbations of HIF1A (underexpressed, unchanged, overexpressed) with Iggy.

In Figure 2.8, we then compared these new ATP productions.

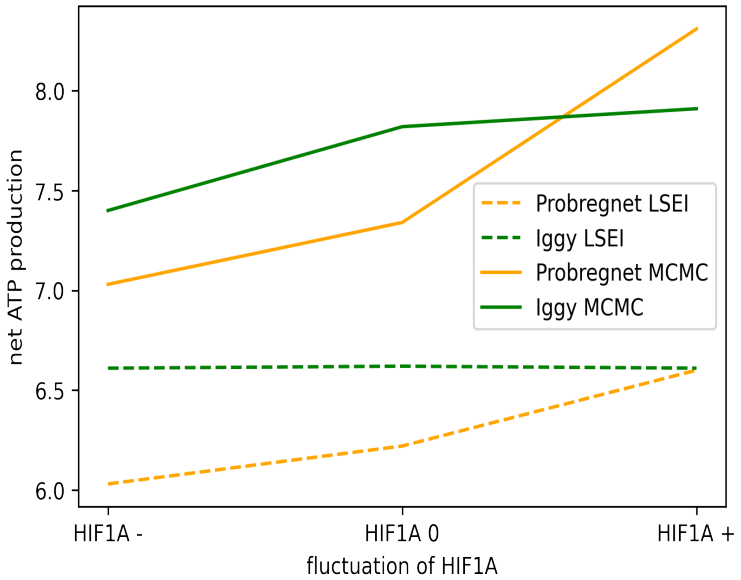


Figure 2.8 – **Impact of Iggy prediction with HIF1A perturbation over net ATP production compared to Probregnet prediction.** In orange, Probregnet enzymes' predictions are integrated with the metabolic network, and in green, Iggy enzymes' predictions are quantified using the observed thresholds before being integrated with the metabolic network. In the dashed line, the LSEI paradigm is applied; in the solid line, the MCMC paradigm is applied.

We concluded that Iggy, unlike Probregnet, did not significantly show the repercussions

of HIF1A perturbation on net ATP production. This observation is all the more true when using the LSEI paradigm rather than the MCMC one. This further highlights the lack of sensibility (to perturbation) we face with Iggy's qualitative predictions.

2.4 Discussion and conclusion

In this chapter we compared two different modelling approaches, Iggy and Probregnet, on two datasets. These approaches perform enzyme *in silico* predictions, upon network stimulation. Both require a prior regulatory network: directed acyclic graph for Probregnet, and directed and signed graph for Iggy; and few experimental samples: 2 samples in two different conditions for Iggy and at least 10 samples in one condition for Probregnet to parametrize the BN.

These methods are intrinsically different in the way their predictions are obtained. Iggy, models network structure and experimental dataset as facts in a logic program that when executed decides if these information is consistent, performs repairs to the data, and when consistent, deduces *coloring models* (solutions) that explain the qualitative signs (or shifts-of-expression) in some nodes of the graph, given a graph topology and an initial set of observations describing a shift of equilibrium (two conditions comparison). Whereas, Probregnet is a two step process : (i) it learns the Bayesian network parameters from a graph topology and multiple experimental datasets, (ii) it computes a belief propagation to predict the quantitative outcome of a system perturbation. We chose these methods since we want to investigate the benefit of a discrete and logical approach, such as Iggy, on the context of gene regulatory and metabolic network integration.

Our results on the Microarray dataset were that Iggy and Probregnet showed very similar (73.3% of agreement) computational enzymes predictions upon the same perturbation. On the second dataset, we obtained different enzyme predictions (only 66.6% of agreement) using both modelling approaches; however Iggy's predictions followed experimentally measured results on enzyme expression. Moreover, concerning other network species, Iggy was more in agreement with experimental observations (65%) than Probregnet ($\approx$ 44%). The lack of a sufficient number (>10) of gene expression profiles (or datasets) in the case of the HUVECs data may have impacted the wrong prediction of Probregnet. As in the first case study, some of the wrong predictions were concerning inhibited enzymes.

Both approaches have their advantages and inconveniences. Probregnet, does not need

a relative (or differential) dataset under another condition. It needs, however, a small network (tens of components). Iggy handles large-scale networks and it has proven its efficiency on networks with more than a thousand of nodes [59]. Interestingly, the integration process between networks and datasets differs for both approaches. Probregnet performs a linear regression of the datasets and requires a previous order (acyclic condition) of the network edges. In comparison, Iggy does not impose this acyclic condition. However, Iggy raises places (data-points) in the dataset where the observation does not agree with the network structure and proposes automatic repairs. In this context, Iggy performs less pre-treatment on the network structure. Furthermore, the nature of the computational predictions of both approaches is different. Iggy predicts a discrete tendency (sign) for the unobserved nodes and not a precise quantitative measure as given by Probregnet. However, Iggy is able to take into account different natures of biological interactions such as complex-formations (modeled with a Boolean *and* gate), activations, or inhibitions. Regarding the computation-time, a test was made for this case-study with a network of more than 4000 edges and 1000 nodes where Iggy's analysis finished in 0.47 s, while Probregnet, after 2 h. This lower computation-time allows Iggy to run several benchmarks of *in silico* perturbations.

As our final aim is to provide an integration of these predictions as constraints in the metabolic reaction equations. Iggy 's discrete predictions are not easy quantifiable (see Section 2.3.3) for this reason we propose a new logical approach based on Iggy logic which allows us to have a finer quantification. This approach called MajS is introduce in the following Chapter 3.

Chapter 3

Second contribution : Predicting weighted unobserved nodes in a regulatory network using Answer Set Programming

Summary of chapter 3

In this Chapter, we focus on improving the modelling of the regulatory network in order to later integrate it with the metabolic network. Previous proposed methods that study this problem fail on dealing with a real-size regulatory network, on computing predictions which are sensible to a perturbation, and on quantifying in a finer way the predicted species behavior. To address previously mentioned limitations, we develop a new method based on Answer Set Programming, MajS. MajS tests the consistency between the input data, proposes minimal repairs on the network to restore consistency, and finally computes weighted and signed predictions over the network species. We tested MajS by confronting the HIF-1 signaling pathway with two gene-expression datasets. Our results show that MajS can predict 100% of unobserved species. When comparing MajS with two tools, one discrete and one quantitative, we observed that compared with the discrete tool, MajS proposes a better coverage of the non-observed species, is more sensitive to system perturbations, and proposes predictions closer to real data. Compared to the quantitative tool, MajS provides finer discrete predictions that go in agreement with the dynamic proposed by the quantitative tool.

3.1 Introduction

In Chapter 2, we focused on the modelling of a regulatory network of the HIF-1 signaling pathway also called Hypoxia signaling pathway, which is of great interest in neurodegenerative diseases. We compared this regulatory network with Alzheimer's disease gene expression data and we perturbed the system by inducing or repressing the HIF1A protein *in silico*. In order to allow us to predict the behaviour of unobserved species, the system was modelled using a logical and a Bayesian approach. We demonstrated that the logical approach, Iggy [8] (see Section 1.4.1), was fast and reliable enough to predict unobserved nodes in the network upon system perturbation when compared to the Bayesian approach, Probregnet [48] (see Section 2.2.3). We have encountered, however, two issues that complicate the regulatory-metabolic network integration process. First, a quantification of Iggy's qualitative predictions (in a three value domain) may introduce new biases to the entire modelling process. Second, because of the semantic of the *sign consistency* underlying Iggy's modelling approach, the comparison uses relaxed rules that do not allow us to distinguish the computational predictions output from two types of *in silico* perturbations in this case-study.

We propose a novel logical approach using Answer Set Programming (ASP), named *MajS*, which addresses the previously mentioned difficulties. This approach, similar to Iggy, compares a regulatory network with gene-expression datasets, searches for inconsistencies, proposes minimal repairs and can predict unobserved nodes. It relies, however, on a different sign-consistency rule which takes into account the majoritarian sign of the nodes' direct predecessors. As an output, added to the consistent sign of a node, it proposes weights which represent the confidence of the predicted sign. We are therefore able to more finely quantify the unobserved nodes. Also, because of the new semantic imposed, we are able to provide predictions more sensitive to the system perturbations. Furthermore, the predictions associated with their confidence weights provide new quantitative insights that make it possible to connect regulatory and metabolic models. Notice that this connection has not been explored in this study.

Our results show that MajS is more stable than Iggy concerning the coverage (the percentage of the number of predicted nodes against all unobserved nodes) of its predictions. In all our performed benchmarks Iggy's coverage fluctuates between $20\% - 100\%$ while MajS is always 100%. Besides, MajS' predictions are more sensitive to perturbation than Iggy's. Indeed, for one of our benchmarks, Iggy outputs the same predicted sign

upon different perturbations whereas MajS allows measuring the change of perturbation on predicted sign, thanks to the notion of weight. We also show that MajS has better accuracy of its predictions compared to *in vitro* perturbed data. Finally, MajS' predictions' dynamic trend agrees with the Bayesian approach predictions.

3.2 Methods

3.2.1 MajS principle

MajS requires as input data an interaction graph (IG), whose edges are directed and labelled as *activation* or *inhibition*. It also requires a list of discrete observations on some IG nodes. This list is composed of discrete values (*colours* or signs) assigned to some of the IG nodes. These values measure the change-of-state of a graph's node (gene or protein) between two specific conditions (e.g. 2 samples corresponding to 2 different biological conditions). The type of discrete assignments provided in the list of observations is: "+" (green) if the node is over-expressed, "-" (red) if under-expressed, and "0" (blue) if there is no change of expression between the two conditions. Not all the graph nodes are included in this list of observations.

We aim at predicting the sign ("-", "0", "+") and weight (a score of confidence of the predicted sign between 0, for low confidence and 100, for high confidence) of unobserved nodes of the IG after it is compared to the list of observations. Prediction can be only computed in case of consistency between the IG topology and the gene expression data measurements. In order to establish consistency, we search for minimal repairs in the IG by adding artificial nodes to the graph. The number of minimal repairs is controlled by a third input K of our method. Our method workflow is detailed in Figure 3.1. The following subsections aim at presenting in detail all the steps of the workflow. Besides, these subsections will present some rules implemented in MajS's ASP code.

Weighted labelling

We propose the two following definition to clarify the next MajS steps.

Definition 3.1 (weighted labelling) A *weighted labelling* is defined as an operation which equips each node of an interaction graph $G = (V, E, \sigma)$ with a sign and a weight associated to this sign. Formally, a weighted label (μ, ω) on a set of nodes $U \subset V$ is defined as a function $U \to \{" - ", "0", " + "\} \times [0, 100]$, where $\mu(v)$ is a function assigning

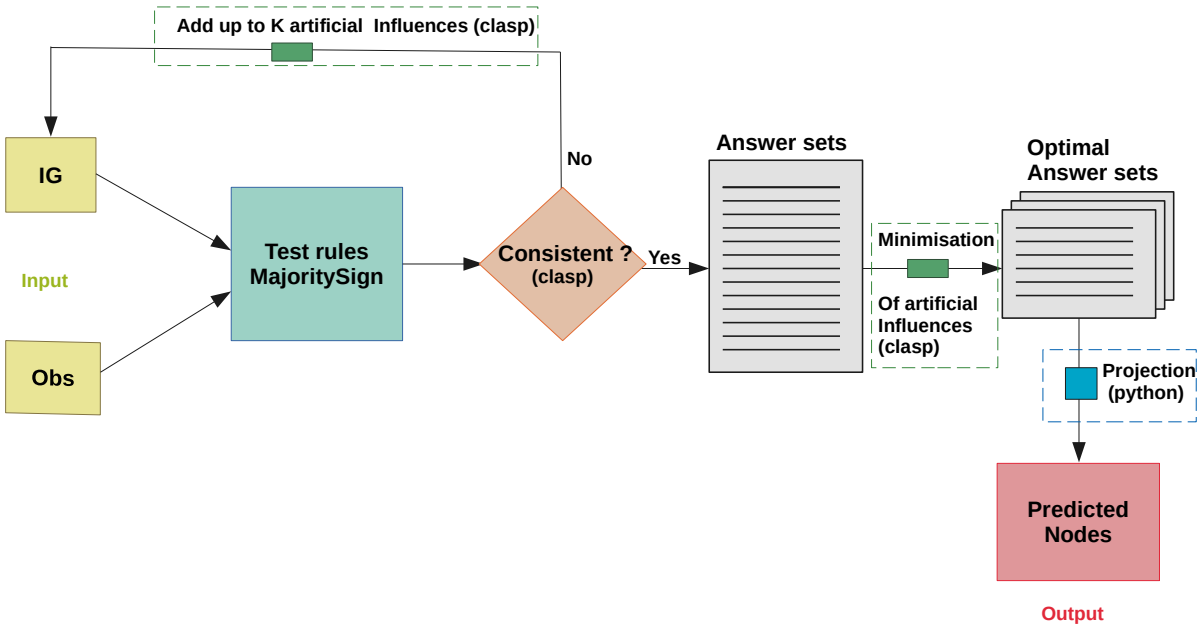


Figure 3.1 – **MajS workflow**. In light green, we show the input data: the interaction graph, **IG**, and the discrete observation list, **Obs**. Then, we apply the **logical rules** implemented in MajS. We test the **consistency** and in case of inconsistency, we **add artificial influences** using K as a fixed parameter. That way we obtain **answer sets** that respect the logical rules. We minimise the artificial influences added to these answer sets and obtain an optimal subset of them. Finally, we project the **optimal answer sets** to obtain as output the **predicted nodes** of our model. clasp is the Answer Set Programming solver [60] used to implement most of MajS steps. Only the *projection* step was implemented in Python.

a *sign* to a node v in U, and $\omega(v)$ is a weight expressing the confidence of the sign.

More precisely, $\mu(v)$ can take three different values: "+" for over-expressed nodes compared to an initial condition; "-", for under-expressed; and "0", for unchanged nodes. Additionally, $\omega(v)$ varies between low confidence (0) and high confidence (100). A weighted labelling is said to be *complete* when it provides a weighted label to each node in V (*i.e.*, $U = V$). In MajS' implementation, the signs are integer values set to -1 for "-"; 1 for "+" and 0 for "0" which allows the use of arithmetic operations on the sign values. Moreover, *Weighted label* is a key predicate inside MajS implementation.

MajS input data

Interaction graph (IG) An interaction graph is defined by a 3-tuple (V, E, σ) where V is a set of nodes, E $\subset \{V \times V\}$ is the set of oriented edges and $\sigma : E \to \{+, -\}$ is a function of the edges where the plus sign represents an activation, and the minus sign represents an inhibition.

Experimental observation (Obs) A list of discrete observations where signs of some IG nodes are given by experimental measure. Generally, a pre-processing step of the experimental data by fixing thresholds for significant expression is required at this point. After the discretisation process, the observed nodes can take three different values: "+" for over-expressed nodes; "-" for under-expressed; and "0" for unchanged nodes.

These experimentally observed nodes belong to a set denoted S. In this study, we fix the weight of all experimental observation nodes to 100 which is the weight representing the maximal confidence.

Test rules MajoritySign

In the following section, we make explicit the logical rules that are applied on the IG and the discrete observation list to test consistency.

1. **Experimental observation signs are kept:** We impose that the sign {" − ", "0", " + "} of the experimental observations in S are kept.
   This is implemented in ASP as follows:

```
sign(-1;0;1).%predicate sign of arity 1, can take 3 values -1 for
    "-"; 1 for "+" and 0 for "0".
weight(0..100).%predicate weight of arity 1, can take a value
    from 0 to 100.
1{weightedLabel(I,S,W): sign(S),weight(W)}1 :- observedNode(I,S).
```

   This line stipulates that an observed node I will keep its sign S, when labelled by MajS represented by the predicate *weightedLabel* of arity 3.

2. **Signed majority wins:** A node is signed "+" or "-", following the majority sign from all its received influences in {"-","+"}. This is implemented in ASP as follows:

```
signMaj(I,S1) :- node(I), countSign(I,S1,N1), countSign(I,S2,N2),
    N1>N2,S1!=0,S2!=0.
```

The predicate *countSign* of arity 3, returns the number (Nx) of received influences over a node I associated with a sign (Sx) which is either "+" or "-". The predicate *signMaj* returns for a node I its majoritarian sign.

3. **Balanced:** A node is signed "0", either if it only receives 0-influences or if it receives the same proportion of signed { "-", "+"} influences. This is implemented in ASP as follows:

```
1 weightedLabel(I,0,100) :- node(I); not signMaj(I,_),
    countSign(I,1,N),countSign(I,-1,N), countSign(I,0,X).
2
3 weightedLabel(I,0,100) :- node(I); not
    signMaj(I,_),countSign(I,0,N),N=#count{P:parent(P,I,_)}.
```

Line 1 stipulates that if both predicates countSign(I,1,N) and countSign(I,-1,N) share the same number of received influences, N then node I is signed as "0" with an associated weight of 100. Line 2 stipulates that for a node I if all the received influences are "0" influences (number N of 0-influences equal to the number of received influences over I), it will be signed as "0" with an associated weight of 100.

4. **Weight assignment:** Every node v of the graph is associated with a sign and a weight, which represents the score on its sign as follows:
   — If $v \in S$ (experimental observations), its weight is fixed to 100.
   — If v is inconsistent and has been repaired, then its weight is fixed to 0.

   ```
   1 weightedLabel(I,S,0) :-  weightedLabel(I,S,_); repaired(I).
   ```

   If a node is repaired, then its weight is fixed to 0.
   — If v is consistent, then the weight is the ratio between the sum of the parent's weights, holding the majoritarian sign, and its number of parents.

   ```
   1 weightedLabel(I,S,W) :- signMaj(I,S); sumWeight(I,S,Z) ;
       C=#count{P:parent(P,I,_)};W=N/C;S!=0; not repaired(I).
   ```

   The *sumWeight* predicate of arity 3 gives for a node I and a sign S, the total weight, Z associated with the sign S received on the node, I. If node I is consistent, not repaired, and has a majoritarian sign. Then, its weight equals the ratio between the number of parents holding the majoritarian sign and the number of parents in total.

The sign-weight couple is denoted as a weighted label. Experimental observations can also be inconsistent after applying MajS rules.

influence Given an interaction graph $G = (V, E, \sigma)$ and a node with a sign μ and a weight ω, for each edge of G (s, v), we define an influence $I(s, v)$ by:

$$I(s, v) = \sigma(s, v)\mu(s).$$

An influence is a 0-$influence$ if and only if $I(s, v) = 0$. It is a positive influence if $I(s, v) = 1$ and a negative one if $I(s, v) = -1$.

Consistency and repairs

A graph is consistent if all of its nodes are consistent. A node is said to be consistent if its weighted label (μ, ω) is in adequacy either with its experimentally observed sign or with the logical rules application, *i.e.*, the *signed-majority*, the *balanced*, and the *weight assignment* (rules 2, 3 and 4 in Section 3.2.1). In case of inconsistency, MajS can repair the graph if its consistency can be established by adding artificial influences. A node is K-repairable when it was inconsistent and became consistent after adding at most K influences. A graph is K-consistent when all the nodes are at least K-repairable. Therefore, the problem is to determine, given an interaction graph G, an experimental observation set S, and an integer parameter K, if G is K-consistent. If this is the case, the minimal sets of repairs to establish consistency are identified. If not, the logical program is unsatisfiable. In Figure 3.2 we show some examples of repair of the inconsistencies on node B.

From answer sets to optimal answer sets

After applying the logical rules and adding repairs in case of inconsistency, we obtain all the answer sets that respect the logical rules. These answer sets are a reduction of the *possible complete weighted labelling* presented in Section 3.2.2. However, these solutions are not optimal as we did not minimise the number of possible repairs. Thus, an optimisation constraint is added to minimise the number of repairs. This constraint respects the logical rules in Section 3.2.1 so that it is guaranteed to find the minimal repairs to establish consistency. This optimisation constraint is implemented as follows:

```
1 #minimize{1,(X,I)  : artInfluence(X,I).}.
```

For each node of the graph (I) we minimise the number of artificial influences (X) added.

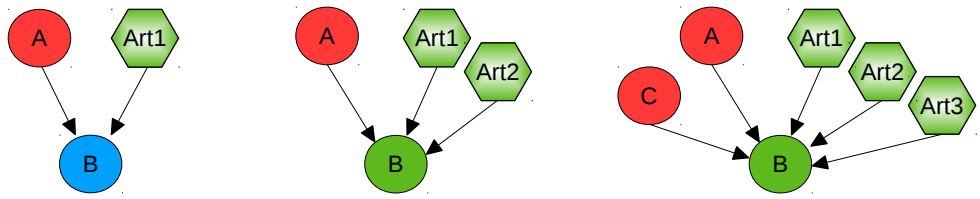


Figure 3.2 – **Inconsistencies between network topology and data**. A species, gene, or protein, is represented as a node. The nodes' colours represent the node's sign when comparing two conditions. They are: blue, "0", or no-change; red, "-", or decrease; and green, "+", or increase. The octagonal nodes represent the artificial influences added by MajS to restore conistency. For Figure a) an artificial influence, noted Art1 is added on node B with the sign "+" to balance with the sign of B's successor node A. For Figure b), two artificial influences, Art1 and Art2, are added to respect the majoritarian sign-consistency rule. Same logic for Figure c).

Predicted nodes obtained after projection

After the optimisation step, many optimal (minimally repaired) answer sets can be proposed. All these solutions are consistent with the logical rules. In order to summarise these results, we add a step called projection. After this step, a node is assigned the following values computed after exploring all optimal solutions: a majoritarian sign (not necessarily unique), the average weight associated with the majoritarian sign, and the standard deviation of the weight. This triplet of values, assigned to all graph nodes, corresponds to the MajS predictions. Finally, MajS takes into account the added repairs in the weight assigned at this step: a node with a 0-weight associated with its sign implies it has been repaired.

The prediction can be either a strong or weak prediction; a strong prediction node means that its sign remains the same across all optimal answer sets and a weak prediction node means that its sign varies.

MajS application on toy example

This section presents the results obtained while applying MajS on a toy example, an IG composed of 10 nodes, 7 activation edges, and 1 inhibition edge ($E \dashv D$), the same toy example used in Section 1.4.1. In Figure 3.3 we illustrate how MajS proceeds when comparing this toy IG with one dataset of observations. First, MajS adds two artificial

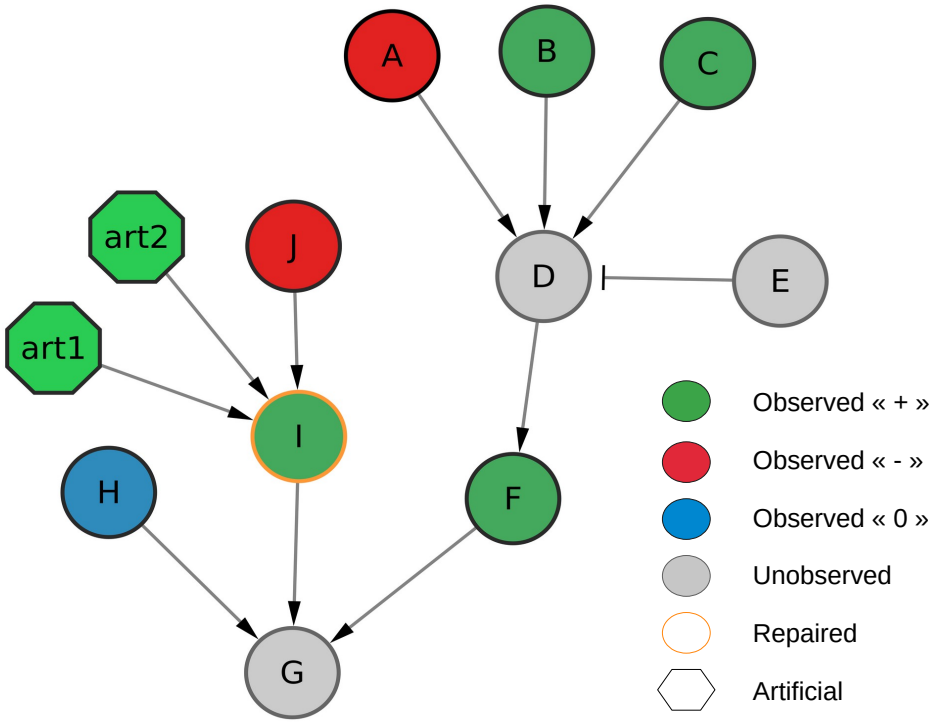
influences (art1 and art2) on node I to establish consistency. Then, it predicts values over nodes D, E, and G. In Figure 3.3, we show the prediction of MajS for nodes D, E, G, and the repaired node, I. This toy example outputs two optimal answer sets: *Solutions 1* and *2*.

Focusing on node D, we observe that in the answer set *Solution 1*, the predicted weight is 75, while in *Solution 2*, the predicted weight is 50. Node D has 4 parents: 3 observed (A ="-", B ="+", C ="+"), and an unobserved parent E. To comply with the *Signed majority wins* rule (see Section 3.2.1), E can be assigned either to "-" or "0". Both assignments give a majoritarian "+" sign on D. When E is set to "-", the weight of the "+" sign on D is 75 as it is defined as the percentage ratio between the sum of the positive influences and the total number of parents of D. A similar reasoning when E is set to "0" leads to the weight of 50 assigned to the "+" sign for D. Notice that no answer set proposes an assignment of E to "+". If that was the case, D's sign would be "0" (*Balanced* rule in Section 3.2.1) and would not explain the sign of its direct successor F; this assignment requires adding another repair and would not be an optimal solution anymore.

To illustrate how projections work (see Section 3.2.1), let us focus on nodes D and E. For node D, the sign across all optimal solutions is "+" so the majoritarian sign given by the projection computation is of "+" (Figure 3.3 (b), column *Projection*, left sub-column *SignMaj*). We also show a detailed view of how this majoritarian sign is represented across all optimal answer sets. First, the number of optimal answer sets having the majoritarian sign for node D is 2. Second, the average weight associated with this sign for D is 62.5. Third, the standard deviation of the average weight is 17. These three values appear represented as a triplet (Figure 3.3 (b), column *Projection*, right sub-column). Following the same logic, for node E, we have two different majoritarian signs: "0" and "-", equally distributed across all optimal answer sets, both average weights are 100, and there is no standard deviation.

Finally, in Figure 3.3 (b), we see that for node I, the weight is fixed to 0 by MajS in all optimal answer sets, implying that it was repaired.

The inputs of the logical program for the toy case study can be found on GitHub: `https://github.com/soph-lebars/MajS/tree/main/toycasestudy`



(a)

Node	Solution 1		Solution 2		Projection	
	μ	ω	μ	ω	SignMaj	[N(SignMaj),mean(ω),sd(ω)]
D	+	75	+	50	+	[2,62.5, 17]
E	-	100	0	100	0 / -	[1,100,0] / [1,100,0]
G	+	66	+	66	+	[2,66,0]
I	+	0	+	0	+	[2,0,0]

(b)

Figure 3.3 – **Toy case study**. (a) Toy network with 7 nodes that are initially observed and 3 unobserved nodes. The I node is marked as inconsistent. (b) MajS predictions on toy network example. All unobserved nodes (grey) are predicted by MajS with a sign (μ) and a weight (ω). The orange node is repaired by adding two artificial influences. Columns Solution 1 and 2 represent sign and weight in optimal answer sets for unobserved and repaired nodes. Column Projection is summarizing all Solution columns as explained in Section 3.2.1.

3.2.2 MajS search space

After having the problem defined, we generate the *choice rules* constructs of our logic program. These rules define the solution space of a problem and create the candidates that are later filtered with the constraints.

Possible complete weighted labellings A complete weighted labelling is composed by both a sign function $\mu : V \to \{\text{“}-\text{”}, \text{“0”}, \text{“}+\text{”}\}$ and a weight function $\omega : V \to [0, 100]$. We also assume that the interval $[0, 100]$ is discretised by the set of integers $\{0, \ldots, 100\}$, in that case, the weight is simply rounded to the closest integer. Thus, the total number of possible complete weighted labelling is equal to $3^{|V|} \times 101^{|V|}$.

Possible repairs Recall that K is a parameter given as input to the method. For each inconsistent node $v \in V$ we generate multiple sets of k artificial nodes and influences, with $k \leq K$. Let us name this set as $p^{(v,k)} = \{p_1, \ldots, p_k\}$. The search space P is defined by the union of all possible ways to assign parents for each node in V, that is:

$$P = \bigcup_{v \in V, 1 \leq k \leq K} p^{(v,k)} \tag{3.1}$$

An artificial parent p_i interacts with v with a positive or negative influence, *i.e.*, $I(p_i, v) \in \{-1, 1\}$. Each p_i is added to the graph G and its influence changes the computation of the majoritarian sign for node v (see rule 2 in Section 3.2.1).

3.2.3 Different and common points between MajS and Iggy

We summarise the main differences and similarities between MajS and Iggy in Table 3.1.

The common points are that the two tools use the ASP paradigm to describe the inputs of the logical program, also called an instance, and to encode the logical constraints of the problem. Secondly, the optimisation of the problem is the same. Both methods seek to minimise the number of influences added to repair.

The two methods are different in three ways. First, the solution space for MajS will be larger because the sign of the nodes is also associated with a weight ranging from 0 to 100. The second difference is that Iggy adds only one influence per node to fix inconsistencies, whereas MajS adds several influences to reestablish consistency. The third difference is in

the logical rules, which are implemented differently. Indeed, Iggy constrains the solution space using sign consistency: the sign of a node is consistent if it can be explained by at least one influence. Whereas, for MajS, the sign of a node is explained by the majority of influences received. In addition, MajS also uses logic rules to constrain the weight. The last difference is in the projection step, which allows obtaining the nodes predicted by MajS and Iggy by summarising the complete list of optimal answer-sets into predictions. For Iggy, the nodes are predicted according to the six values shown in Table 3.1. MajS summarises the optimal answer sets by predicting the nodes with their majoritarian sign, their associated weight and the associated standard deviation.

3.2.4 Comparison of discrete predictions with continuous values

Here, we describe our method for comparing the discrete predictions provided by MajS and Iggy to the continuous experimental values of fold change. We rely on mixtures of normal distribution which is proven to be a probabilistic model of choice for microarray experiments [61]. For each predicted node, we define a continuous probability distribution $\mathcal{M}$, whose density function is denoted as $M(x)$ with $x \in \mathbb{R}$, that is a mixture of three Normal distributions whose means depend on the sign of the prediction and whose standard deviations depend on the weight of the prediction. Precisely, for $x \in \mathbb{R}$ the mixture density function is defined by

$$M(x) = \sum_{s \in \{-,0,+\}} \phi_s \cdot N_s(x), \tag{3.2}$$

where ϕ_s is the ratio of answer sets for which the node sign is predicted as s for $s \in \{-, 0, +\}$, and $N_s(x)$ is the density function of a Normal distribution $\mathcal{N}(\mu_s, \sigma_s)$ with mean μ_s and standard deviation σ_s. Precisely, one has

$$N_s(x) = \frac{1}{\sigma_s \sqrt{2\pi}} \, \mathrm{e}^{-\frac{1}{2}\left(\frac{x-\mu_s}{\sigma_s}\right)^2}.$$

In our comparison, we fix mean values using experimental HUVECS data (see Section 3.3.2 for more details). The standard deviation is derived from the weight $w \in [0, 100]$ of the sign s for a given node by using a simple linear transformation rule $\sigma_{hc} \cdot \frac{w}{100} + \sigma_{lc} \cdot (1 - \frac{w}{100})$. Here, σ_{lc} is a fixed constant considered as a low confidence prediction (which is assigned when $w = 0$) and σ_{lc} is a fixed constant considered as a high confidence prediction (which is assigned when $w = 100$).

Table 3.1 – Different and common points between Iggy and MajS

	Iggy	**MajS**
Instance (input)	• An interaction graph, whose edges are directed and labelled as *activation* or *inhibition*. • A list of discrete observations on some IG nodes.	• An interaction graph, whose edges are directed and labelled as *activation* or *inhibition*. • A list of discrete observations on some IG nodes.
Search space/ Guess	• Depends on node, sign $\in \{\text{“}-\text{”}, \text{“0”}, \text{“}+\text{”}\}$ $(3^{\lvert V\rvert})$ • 1-influence repair added by node	• Depends on node, sign $\in \{\text{“}-\text{”}, \text{“0”}, \text{“}+\text{”}\}$, weight $\in [0, 100]$ $(3^{\lvert V\rvert} \times 101^{\lvert V\rvert})$ • K-influences repair added by node
Logical rules	• Experimental observation signs are kept. • A node signed as “0” must receive only one influence signed as “0” or at least one “+” and one “−” influence. • A signed node must be justified by at least one signed influence.	• Experimental observation signs are kept. • A node is signed “0” either if it only receives 0-influences or the same proportion of signed “−”, “+” influences. • A node is signed following the majority sign from all its received influences.
Optimisation	• Minimise the number of added repairs	• Minimise the number of added repairs
Projection (predicted nodes)	• Six levels of possible prediction: 1 - 2 notPlus (-, 0) 3 0 4 notMinus (0, +) 5 + 6 CHANGE (+, -)	• Majoritarian sign • Statistical information on the weight distribution (average, standard deviation)

To compare MajS and Iggy's methods, once we have a (mixture) density function $M(x)$, we calculate:

$$P(fc) = Prob\{|F - fc| < \varepsilon\} = \int_{fc-\varepsilon}^{fc+\varepsilon} M(x)dx, \tag{3.3}$$

where ε is fixed to 0.005.

In order to improve the significance of $P(fc)$, we compute the maximum value that can be reached by any mixture obtained within these settings. It is straightforward that the mixture that provides the maximum value is the one corresponding to a single prediction, say "0", with weight 100. The maximum for $P(x)$ is then reached for $x = 0$. Consequently, the maximum value, denoted as $\mathcal{P}_{max}$ equals :

$$\mathcal{P}_{max} = \int_{-\varepsilon}^{+\varepsilon} \frac{1}{\sigma_{hc}\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\left(\frac{x}{\sigma_{hc}}\right)^2} dx = 0.07969$$

when $\varepsilon = 0.05$ and $\sigma_{hc} = 0.05$.

Finally, we define a significance score $\mathcal{S}(fc)$ between 0 and 1:

$$\mathcal{S}(fc) = P(fc)/\mathcal{P}_{max} \tag{3.4}$$

We use this significance score to compare both Iggy and MajS methods.

Our method for computing a significant score relies on a few number of parameters. The first one is the ε parameter, used to compute the area under the curve of the distribution. We also use two parameters (σ_{hc} which stand for high confidence and σ_{lc} which stand for low confidence) in order to transform the weight given by MajS into a standard deviation involved in the normal distribution calculation. To observe the impact of these arbitrary choices on the significance scores and the comparison of MajS and Iggy methods, we make some tests with different values of these parameters. All the experiments are available on the GitHub repository.

For the epsilon parameter (ε), we lead some experiments to observe the impact of this arbitrary choice on the conclusion we make regarding the comparison of scores of both methods MajS and Iggy. We define a threshold $\mathcal{E}$ with $\varepsilon \leq \mathcal{E}$ where, for all values of ε, no change is observed with respect to the conclusion of the comparison of both methods (*i.e.,* a score of a method become better than the other). We test different values of $\varepsilon = \{0.001, 0.002, ..., 0.1\}$ in order to find a $\mathcal{E}$ adapted to our data. This experiment concludes that only two genes have different behaviour. Finally, we can deduce a value of $\mathcal{E} = 0.015$. In our study, we take $\varepsilon = 0.005$ that is far below $\mathcal{E}$. Notice that it is not the choice of epsilon that matters, but the conclusion of the comparison of scores of the two methods.

For the Low and high confidence parameters, respectively σ_{hc} and σ_{hc}. These two parameters are fixed for the weight transformation into the standard deviation of the

normal distribution. The objective of this transformation is to attribute a small standard deviation value when the weight is high, referring to a high confidence in the prediction. In contrast, the weight is transformed to a high value when it represents a low confidence in the prediction. In this case, the distribution mixtures are flattened. We test different values of high confidence ($\sigma_{hc} = \{0.01, 0.02, ..., 0.1\}$) and low confidence ($\sigma_{lc} = \{0.1, 0.2, ..., 1\}$) parameters. Like the preceding experimentation, we look at the difference in the scores of the both methods. We can identify some combinations of parameters values where no change is observed (Figure 3.4). Indeed, we deduce an interval I for σ_{hc} and σ_{lc} where the number of genes remains constant, thus leading to the same conclusions when comparing Iggy and MajS: $I_{\sigma_{hc}} = [0.05, 0.1]$ and $I_{\sigma_{hc}} = [0.5, 0.8]$. In our study, we fix $\sigma_{hc} = 0.05$ and $\sigma_{lc} = 0.5$.

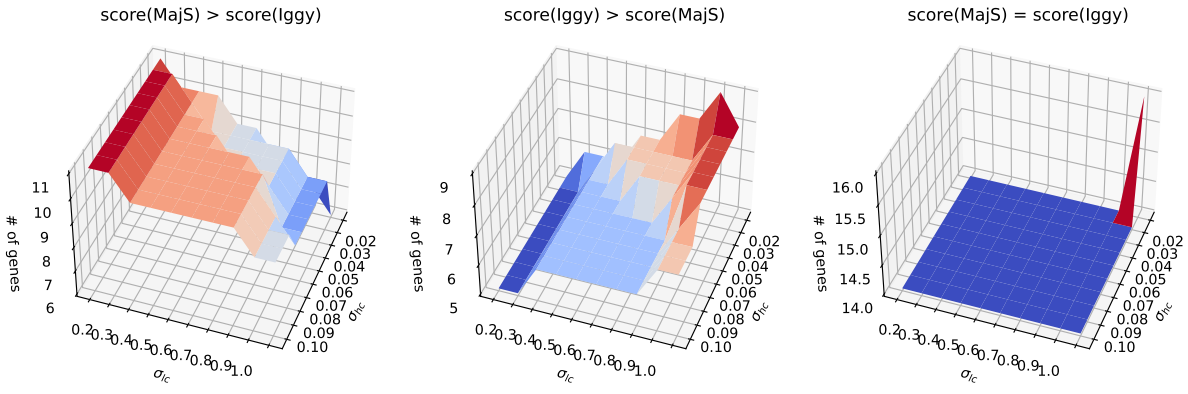


Figure 3.4 – Representation of the number of genes relevant in a specific case (score(MajS)>score(Iggy), score(Iggy)>score(MajS), score(MajS)=score(Iggy)) according to different values of σ_{hc} and σ_{lc}. This experimentation concerns the Benchmark1.

3.3 Results

We show in this section the results obtained after applying MajS on three case-studies. All scripts and data are available on GitHub: `https://github.com/soph-lebars/MajS`.

3.3.1 Case studies

We focus on the regulatory network, modelled by an interaction graph (IG), and the impact of a perturbation on this regulatory network evaluated by discrete observations obtained from two gene expression datasets. These datasets and discrete observations generation are more detailed in Section 2.2.1 of Chapter 2.

Biological network - interaction graph We focus on the regulatory network of the HIF-1 signalling pathway, known to be of importance in neurodegenerative diseases [55]. This graph was extracted from the KEGG database. Nodes represent proteins or genes, and edges represent activations or inhibitions between two nodes. We reduce the regulatory network by keeping only nodes associated with expressed genes in the two datasets used in this study. The two networks, respectively reduced with Alzheimer's disease (AD) and RNA-Seq datasets, are composed of 94 and 81 nodes and 285 and 233 edges.

Datasets We evaluate our model against two datasets composed of gene differential expression in two conditions. The Microarray dataset corresponds to cells from the hippocampus brain region [50]. It compares data from AD patients to data from Healthy individuals. The RNA-Seq dataset corresponds to HUVECS (Human umbilical vein endothelial cells) [53]. It compares the HUVECS response to an induced overexpression of HIF1A to one with a normal HIF1A expression.

Benchmarks We aim to study the impact of perturbing the system with a focus on the node HIF1A, a key protein of the HIF-1 signalling pathway. Recall that, one of the inputs of our method consists of a list of discrete observations for which a significant change of expression is detected between two conditions. The changes of expression our method accepts are: "+", over-expression; "-", under-expression; and "0", no-change of expression. The values of the thresholds used to detect significant over- or under-expression are fixed according to the nature of each dataset as detailed below.

Thresholds choice in HUVECS Benchmarks For the RNA-Seq dataset, we used the logFC (log of gene expression) from cells with HIF1A *in vitro* over-expressed over normally expressed genes that were already provided in [53]. We use a threshold of 1.5 that is commonly used for logFC as said in [62]. The genes with logFC over 1.5 are set to "+", the ones below -1.5 are set to "-", and the ones between -0.15 and 0.15 are set to "0". Using these thresholds, we obtain 30 observed nodes (out of 81 in the graph).

Thresholds choice in AD Benchmarks We aimed to study the impact of perturbation over HIF1A on the enzymes for the AD dataset. We used a threshold over the fold change distribution. The fold change is the expression of the gene in AD patients over the expression of the corresponding gene in Healthy individuals of all the genes in this dataset. The genes with FC that are over the third quartile are set to "+"; the ones under the first quartile are set to "-"; and the ones between 0.99 and 1.01 are set to "0". Using these thresholds, we obtain 64 nodes (out of 94 in the graph) that compose the input observation list of our method. For the AD case, perturbations of HIF1A are only done *in silico*. We generate 3 different perturbations by adding the following observations to the list of 53 observations, described before: (plus) HIF1A='+', (minus) HIF1A='-', and (zero) HIF1A='0'.

3.3.2 MajS applied to model HIF-1 signalling pathway and HUVECS dataset integration

Data

The IG for this case study is composed of 81 nodes and 233 edges derived from the HIF-1 signalling pathway and compared with a RNA-Seq dataset from HUVECS (see Section 3.3.1). This IG is compared with two different lists of discrete observations; denoted by *Benchmark 1* and *Benchmark 2* in Table 3.2. Benchmark 1 is composed of 30 nodes that are a partial observation of the IG, generated by estimating significantly expressed genes in the RNA-Seq dataset using specific thresholds (see Section 3.3.1). Benchmark 2, composed of 25 nodes, is a modification of Benchmark 1; where we have altered or removed the value of 9 observations, direct neighbours of HIF1A or directly linked to the network enzymes. These modifications were done to improve the coverage of Iggy. By modifying these observations, the problem becomes simpler to solve for Iggy, leading to

a better coverage for Iggy. All these benchmarks are available on the GitHub companion repository.

MajS results on HUVECS dataset

On Benchmarks 1 and 2, MajS generates predictions for all initially unobserved nodes. MajS is configured by setting $K = 3$ as the maximum artificial influences per node. The computations took approximately 97 s. [1] for each benchmark. MajS obtains 2016 optimal answer sets for both benchmarks, that is, 2016 assignments of nodes with a sign. The number of minimal artificial influences added by MajS to restore consistency in both benchmarks was 8. They are spread over seven repaired nodes and a maximum of 2 artificial influences per repaired node. The HIF-1 signalling IG is 2-consistent concerning the HUVECS dataset (see Section 3.2.1).

Comparison of MajS and Iggy

The aim of the sections from *Comparison of MajS and Iggy* to *comparison of MajS and Iggy predictions with real data* is to understand the difference in prediction on the HUVECS dataset between MajS and Iggy. Iggy is described in Section 1.4.1 and in [14].

In Table 3.2, we show a global comparison of both tools with the two different benchmarks. For Benchmark 1, 51 nodes are unobserved in the IG. We can see that MajS was able to predict all of them (100% of coverage), whereas Iggy could predict only 30 nodes (59% of coverage). In order to compare the predictions' signs for both methods, we consider for MajS the majoritarian sign of the predicted nodes. 22 nodes are predicted with the same sign for both methods, while 8 nodes are predicted differently between both methods. For Benchmark 2, we obtain for both methods 100% of coverage. The number of predicted nodes in common is 48, and the number of predicted nodes different remains 8. Besides, these 8 nodes are the same for both benchmarks.

The different coverage obtained by MajS and Iggy in Benchmark 1, is explained by the different type of rule imposed to each graph node in both methods. Recall that Iggy implements a sign consistency rule stating that *a node sign has to be explained by at least one signed influence received*, whereas MajS implements a majoritarian sign rule stating that *a node sign has to be explained by the majoritarian sign of the influences received*. When a node receives a positive and a negative influence, Iggy cannot infer any

1. solver: clingo version 5.5.0, parallel execution on 10 cores. All computations are performed on a standard laptop machine. Ubuntu 18.04, 64 bits, intel core i7-9850H CPU 2.60 GHz, 32 GB.

prediction (both “+" and “-" scenarios are possible) whereas MajS will predict either 0, in case of balance, or the majoritarian sign. For that reason, MajS is always generating more predictions than Iggy.

This is illustrated by the coverage comparison in Benchmark 1.

Table 3.2 – Table of comparison between Iggy and MajS for two benchmarks.

	Benchmark 1		**Benchmark 2**	
	MajS	**Iggy**	**MajS**	**Iggy**
Predicted node	51	30	56	56
Coverage of predicted node	100.0 %	59,00 %	100.0 %	100.0 %
Number of predicted node : common VS different	**22 VS 8**		**48 VS 8**	

Different computational predictions for Iggy and MajS

In Table 3.3, we present the eight nodes predicted differently. The six nodes in green are predictions for which there is an intersection between Iggy and MajS predictions. For example: (i) for PLCG1, the predicted sign of MajS, “-", is included in the prediction “notPlus" of Iggy, and (ii) for RBX1, the prediction of Iggy is included in the prediction of MajS. The two orange nodes in Table 3.3 refer to different predictions between Iggy and MajS. However, for these nodes, the number of cases that MajS predicted the same sign as Iggy remains high (672/2016) despite not being majoritarian.

To illustrate this prediction difference between MajS and Iggy, we can analyse PLCG nodes (PLCG1, PLCG2) in detail. MajS summarises all the optimal answer sets, so it outputs the majoritarian sign, its average weight and the standard deviation. For PLCG nodes, MajS gives “-" as the majoritarian sign, but we can see that “0" is also present in the optimal answer sets distribution. Iggy does not allow this distribution analysis and outputs “notPlus", which signifies that there are “-" and “0" in the optimal answer sets but cannot allow determining the most representative one.

To conclude, MajS allows more information on predicted nodes than Iggy, and it outputs predictions that are, for most of the cases, with signs that often coincide with those predicted by Iggy. MajS outputs more detailed information than Iggy: number of answer sets and weight distribution.

Table 3.3 – Table of different predicted nodes between Iggy and MajS. **SignMaj** refers to the majoritarian sign across all optimal answer sets. Columns 3-5 show the detailed distribution for each predicted sign across all optimal answers (in total 2016) sets according to MajS; the numbers in brackets refer to the number of answer sets where the node was fixed to this sign, the average weight and its standard deviation. **SignIggy** refers to the sign predicted by Iggy.

Name	SignMaj	Sign= " + "	Sign= " - "	Sign= " 0 "	SignIggy
PLCG1	-	[0, 0, 0]	[1344, 100, 0]	[672, 100, 0]	notPlus
PLCG2	-	[0, 0, 0]	[1344, 100, 0]	[672, 100, 0]	notPlus
RBX1	0 / -	[504, 100, 0]	[756, 100, 0]	[756, 100, 0]	0
VHL	0 / -	[504, 100, 0]	[756, 100, 0]	[756, 100, 0]	0
RELA	+ / 0	[756, 44, 16]	[504, 44, 16]	[756, 100, 0]	0
NFKB1	+ / 0	[756, 44, 16]	[504, 44, 16]	[756, 100, 0]	0
IFNGR1	+	[756, 100, 0]	[588, 100, 0]	[672, 100, 0]	0
IFNGR2	+	[756, 100, 0]	[588, 100, 0]	[672, 100, 0]	0

Comparison of MajS and Iggy predictions with real data

Using a normal distribution mixture (see Section 3.2.4), we compare the significance score of both methods to predict the real fold change data. This one is extracted from HUVECS dataset (see Section 3.3.1) where the perturbation was conducted *in vitro*. The aim of comparing this data with both methods is to see if they are able to model a perturbation *in silico* and have predicted fold change close to the fold change with an *in vitro* perturbation. This comparison is conducted as a validation.

We apply our method introduced in Section 3.2.4 using $\mu_- = -0.394$, $\mu_0 = 0$ and $\mu_+ = 0.489$ as parameters of the three normal distributions. These values are the respective means observed in the experimental HUVECS data using the thresholds fixed in Section 3.3.1. The standard deviation is calculated using $\sigma_{lc} = 0.5$, which is approximately equal to the difference between two means (e.g., $\mu_+ - \mu_0$ for instance), and $\sigma_{hc} = 0.05$. As an illustration, the computed standard deviation is respectively 0.5 when $w = 0$, 0.05 when $w = 100$, and 0.275 when $w = 50$. Notice finally, that all predictions provided by Iggy are to be considered with a high confidence weight, so they are assumed to have weight $w = 100$ in our comparison.

For Benchmark 1, 21 unobserved genes (out of 51) are left out due to not being predicted by Iggy (see Section 3.3.2). If we focus, for example, on the PLCG1 gene prediction, MajS's mixture provides a higher significance score than Iggy's (Figure 3.5). According

to Equation 3.2, the obtained mixture density function for MajS is

$$M_{MajS}(x) = \frac{1344}{2016}N_{-}(x) + \frac{672}{2016}N_0(x) + \frac{0}{2016}N_{+}(x),$$

with $N_{-}(x)$, $N_0(x)$, and $N_{+}(x)$ being the probability density functions of three normal laws with different means and standard deviations (see Section 3.2.4 for details). Coefficients of these functions are the ratio of answer sets for which the node sign is predicted (see columns 3-5 of the Table 3.3, node PLCG1). The obtained mixture density function for Iggy is

$$M_{Iggy}(x) = \frac{1}{2}N_{-}(x) + \frac{1}{2}N_0(x) + \frac{0}{2}N_{+}(x),$$

with the same density functions $N_{-}(x)$, $N_0(x)$ as for MajS in this example. Given the "not-Plus" predicted sign for PLCG1 node (see SignIggy column in Table 3.3), corresponding to an equivalent prediction of "-" and "0", the ratios for $N_{-}(x)$ and $N_0(x)$ density functions are equal to $\frac{1}{2}$.

According to Equation 3.4 and to predict the log fold change value of PLCG1 (-0.38), the significance scores are equal to 0.62 for MajS and 0.47 for Iggy. Thus, MajS method provides a better prediction for the PLCG1 gene.

Considering the 30 initially unobserved genes of Benchmark 1, we found that MajS and Iggy produce the same mixture density for 14 genes, thus providing the same prediction. However, for the 16 other genes, MajS provides a better prediction for 10 out of 16 genes, in the sense that the computed score for MajS is greater than the one computed for Iggy. The same observations can be done for Benchmark 2 with 56 unobserved nodes; the computed prediction scores are equal for 16 genes; for 33 genes, MajS provides a higher score while Iggy provides a higher score for only 7 genes. This supports that MajS can obtain higher confidence on the predicted signs than Iggy does.

3.3.3 MajS applied to model HIF-1 signalling pathway and Alzheimer's disease (AD) dataset integration

This section focuses on the enzyme prediction to do the link with the metabolic network. Our first aim is to illustrate the difference in coverage and sensitivity between the two discrete approaches: MajS and Iggy, on a Microarray dataset of Alzheimer's Disease patients. Our second aim is to compare MajS with a Bayesian quantitative approach, Probregnet, to point out the similarity of predictions in terms of dynamic evolution across

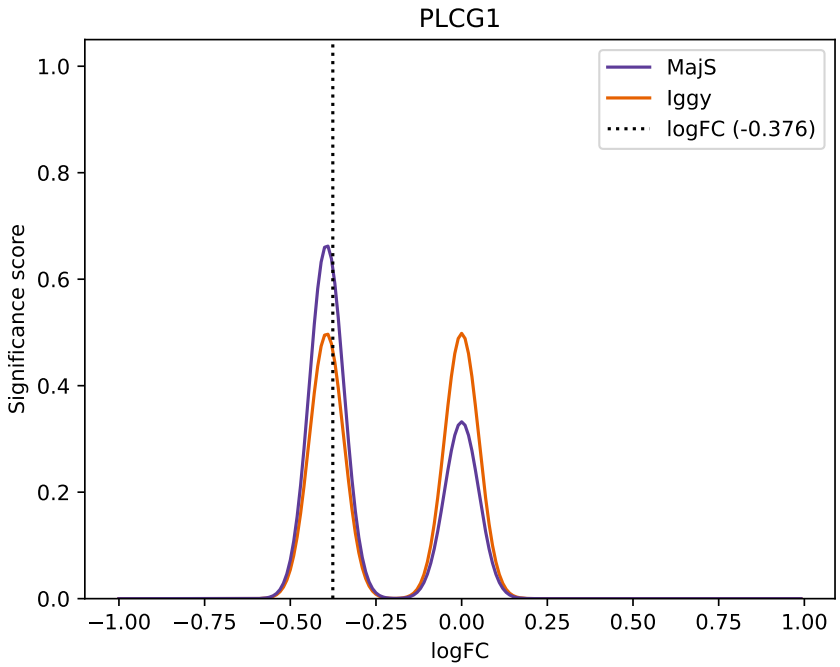


Figure 3.5 – **Iggy's and MajS' mixture of PLCG1 gene.**

different perturbations. Probregnet [48] predicts unobserved nodes with quantitative values in a regulatory network (see details in Section Background 3.2).

Data

The IG for this case study comprises 94 nodes and 283 edges. It was derived from the HIF-1 signalling pathway (see Section 3.3.1) and initially compared to three different lists of discrete observations. Each list is based on the *AD Benchmark* generated by estimating significantly expressed genes in the AD dataset using thresholds (see Section 3.3.1). The three lists of observations were derived by fixing the value of node HIF1A to "+", "-", or "0" in order to simulate *in silico* a HIF1A perturbation. These three *original datasets* were composed of 64 observed nodes. When comparing the IG with these original datasets, Iggy provided new predictions for only 9 nodes (out of 30 unobserved ones), whereas MajS predicted the 30 nodes.

In order to provide a comparison of the predictions of MajS, Iggy and Probregnet for this case study, we modified the original datasets by fixing the value of HIF1A neighbours and removing observations referring to enzymes. In total, we performed 14 modifications.

These modifications were done with a similar idea as for the HUVECS dataset (see Section 4.3.1), to improve Iggy's prediction coverage. With the original dataset, Iggy predicted less than 30% unobserved nodes. Besides, observed enzymes are removed to see if the prediction agrees with the real dataset observations.

Our three resulting lists of observations were composed of 54 observations. We denote these three benchmarks as: *Benchmark_zero*, *Benchmark_plus*, and *Benchmark_minus* referring to HIF1A fixed to "0", "+", "-" respectively. The rest of this section is presented with these modified benchmarks.

MajS results on AD dataset

The computation took less than 40s. for each benchmark. For *Benchmark_minus* we obtain 480 optimal answer sets with 32 repaired nodes, for *Benchmark_zero* we obtain 320 optimal answer sets with 27 repaired nodes and *Benchmark_plus* we obtain 160 optimal answer sets with 28 repaired nodes. The number of artificial influences K added for each repaired node by MajS to restore consistency was maximum 4. The IG is 4-consistent for the AD dataset (see Section 3.3.1).

Difference of coverage between MajS and Iggy across all benchmarks

MajS predictions' coverage is 100% across all three benchmarks (40 unobserved nodes). For Iggy the coverage was of 20%, 88%, and 85% for benchmarks having HIF1A set to "-", "0", and "+", respectively. MajS has better coverage than Iggy for this case study.

MajS is more sensitive than Iggy to nodes perturbations

This section focuses on the 15 enzymes present in our IG; a more refined discrete prediction of these nodes may facilitate the IG model integration with a metabolic network model. Table 3.4 presents the computational predictions of MajS and Iggy on the enzyme nodes when comparing the IG with the three datasets of observations with different values ("-", "0", "+") set for HIF1A. All MajS predictions were strong predictions (*i.e.*, no variation across all optimal answer sets) with a unique weight.

For *Benchmark_minus*, Iggy could not predict the enzyme signs, while MajS was able to give a majoritarian sign of "0" associated with a weight of 100 to all the enzymes. This is explained by the different rule imposed to each node by Iggy (sign-consistency)

and MajS (majoritarian sign). As found for the HUVECS dataset (see Section 3.3.2), Iggy constraints less the problem, generating more answer sets and producing fewer predictions.

For *Benchmark_zero* and *Benchmark_plus*, Iggy proposes similar predictions; consequently, it is not possible to observe any impact of the HIF1A perturbation. Instead, for MajS 12 enzymes are predicted as "+", and they hold different weights (25 and 50 respectively) according to the benchmark. MajS is more sensitive than Iggy to perturbations on node HIF1A for this case study. It allows a measurable repercussion of the perturbation with different strengths for most enzymes. This is possible thanks to the weight term used in the domain of the answer sets obtained with MajS, and the weight assignment rule (see Section 3.2.1).

Table 3.4 – MajS and Iggy predictions upon perturbations of HIF1A for 3 Benchmarks. *Benchmark_minus* contained HIF1A="-"; *Benchmark_zero* contained HIF1A="0" and *Benchmark_plus* contained HIF1A="+". Here, "Na" means that Iggy could not predict for this Benchmark. MajS gives a predicted node as a tuple composed of the majoritarian sign and its average weight; the standard deviation is 0. The colours are focused on Benchmark_zero and Benchmark_plus; the enzymes predicted with "+" sign appear in green. The ones predicted with "-" appear in pink.

	Benchmark_minus		Benchmark_zero		Benchmark_plus	
Name	MajS	Iggy	MajS	Iggy	MajS	Iggy
ALDOA	(0,100)	Na	(+,25)	+	(+,50)	+
ENO1	(0,100)	Na	(+,25)	+	(+,50)	+
ENO2	(0,100)	Na	(+,25)	+	(+,50)	+
ENO3	(0,100)	Na	(+,25)	+	(+,50)	+
GAPDH	(0,100)	Na	(+,25)	+	(+,50)	+
HK1	(0,100)	Na	(+,25)	+	(+,50)	+
HK2	(0,100)	Na	(+,25)	+	(+,50)	+
HK3	(0,100)	Na	(+,25)	+	(+,50)	+
LDHA	(0,100)	Na	(+,25)	+	(+,50)	+
PFKL	(0,100)	Na	(+,25)	+	(+,50)	+
PGK1	(0,100)	Na	(+,25)	+	(+,50)	+
SLC2A1	(0,100)	Na	(+,25)	+	(+,50)	+
PDHA1	(0,100)	Na	(-,100)	-	(-,100)	-
PDHA2	(0,100)	Na	(-,100)	-	(-,100)	-
PDHB	(0,100)	Na	(-,100)	-	(-,100)	-

.

Comparison of MajS and Probregnet evolution of prediction of the 15 enzymes upon HIF1A perturbation

The focus here is on comparing the evolution of MajS' predictions concerning the quantitative predictions of Probregnet [48] for the three types of HIF1A perturbations for the 15 enzymes. The repercussion of the HIF1A perturbation by Probregnet was monitored using the ratio, noted FC (fold change), between the node expression in the perturbed model (expression in AD patients) and the one in a non-perturbed model (expression in Healthy individuals).

Figure 3.6 shows that 4 (out of 15) enzymes have a different evolution when comparing MajS to Probregnet. For MajS, the variation is measured by considering both the sign and the weight. According to MajS, all enzymes tend to increase in the transition from HIF1A="-" to HIF1A="+" except for PDH enzymes which tend to decrease. These variations agree with the IG topology; indeed, all enzymes are activated by HIF1A except PDH enzymes which are indirectly inhibited. According to Probregnet, 10 out of 15 enzymes are increasing (9/10 in agreement with MajS). In the 5 decreasing enzymes, there is a smaller proportion (2/5) of agreement with MajS. 3 out of 4 disagreements correspond to Probregnet enzyme predictions which are not significant, with a delta variation less than 0.02 when the average delta is 0.3 for the rest of the enzymes.

MajS and Probregnet give a similar dynamic trend for most enzyme predictions.

3.3.4 Quantification of MajS' predictions

In this section, we search to quantify MajS enzymes' qualitative predictions. In order to conduct this quantification, we focused on the AD case study and the prediction for the 15 enzymes presented in Table 3.4.

Our method to quantify MajS results relies on the assumption that *Benchmark_zero* in Table 3.4 represents HIF1A without perturbations, so the enzyme Fold-Change (FC^0_{enz}) in *Benchmark_zero* is equal to its Fold-Change in the AD dataset. Precisely, the average enzyme expression for AD patients is denoted as X^{AD}_{enz} and the average enzyme expression for healthy individuals is denoted as X^h_{enz}. Therefore, the (FC^0_{enz}) is defined as:

$$FC^0_{enz} = X^{AD}_{enz} / X^h_{enz} \tag{3.5}$$

From this assumption, we deducted mathematical formulations to compute the enzyme Fold-Change for the other benchmarks using the sign and the weight given by MajS.

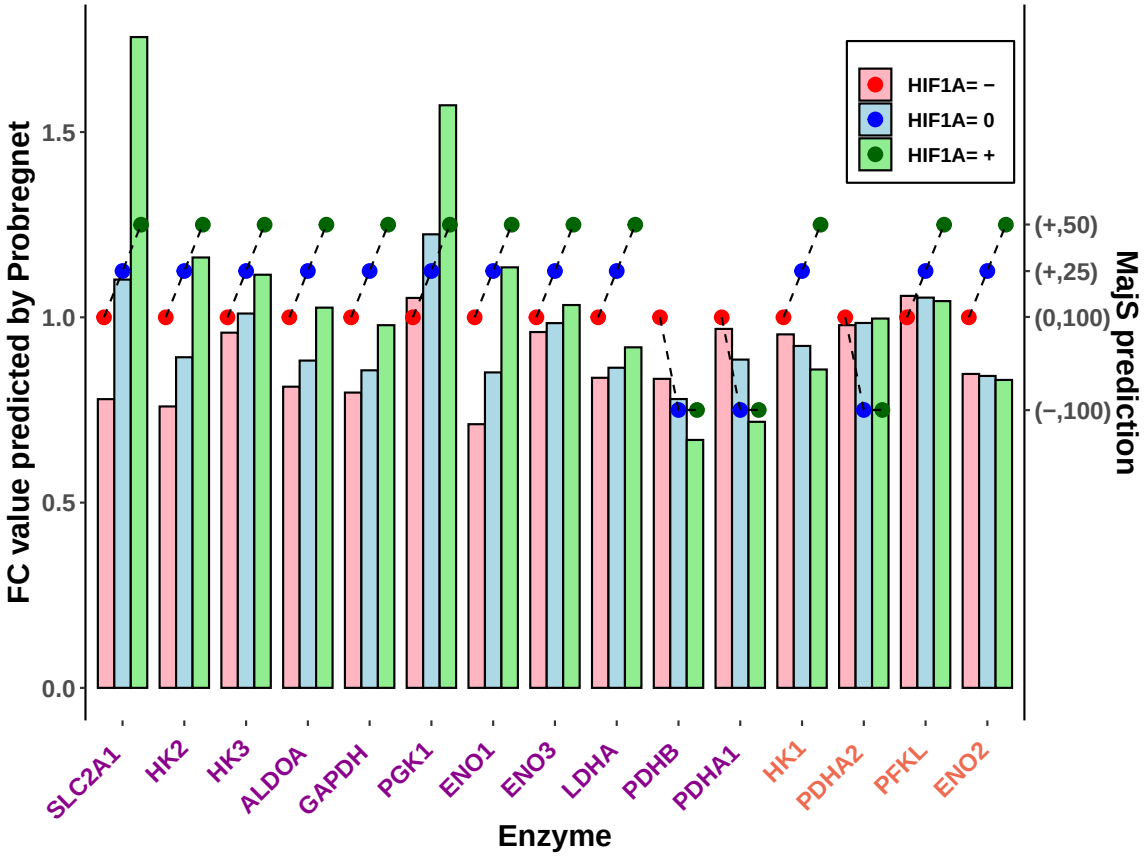


Figure 3.6 – **MajS and Probregnet enzymes predictions.** For MajS, the evolution of prediction across the three HIF1A perturbations is shown with a dashed line and for Probregnet, with three consecutive bars for each enzyme. The left y-axis shows the foldchange (FC) predicted by Probregnet. On the right y-axis is the weighted label given by MajS. The x-axis shows the names of the enzymes. In purple, those that agree on evolution across perturbations between Probregnet and MajS. In orange, those with a disagreement on evolution.

The FC of the enzyme for the *Benchmark_minus* is computed as follows:

$$\begin{aligned} FC^{-}_{enz} &= K^{-} * FC^{0}_{enz} \\ \text{where } K^{-} &= 1 + \Delta^{-} \\ \text{and } \Delta^{-} &= [t(\mu^{-}, \omega^{-}) - t(\mu^{0}, \omega^{0})] \end{aligned} \tag{3.6}$$

K^{-} is a constant associated with the observed benchmark, here *Benchmark_minus*. Δ^{-} is a difference between *Benchmark_minus* observed sign μ^{-} and weight ω^{-} and *Bench-*

mark_zero observed sign μ^0 and weight ω^0 after applying the function t.

In more detail, $t(\mu, \omega)$ is a function which associates to a sign and a weight a value between $[-1, 1]$ such as:

$$t(\mu,\omega) = \begin{cases} 0, & \text{if } \mu = 0, \quad \forall w, \\ -\omega/100, & \text{if } \mu = -, \\ +\omega/100, & \text{if } \mu = + \end{cases}$$

For the PDH enzymes in Equation 3.6, we added a correction term as done in the Probregnet pipeline [48], which allows taking into account the weight of PDK1, a direct inhibitor of PDH enzymes. The correction is performed by multiplying $t(\mu^0, \omega^0)$ with $\omega^0_{PDK1}/100$ which is equal to 0.25.

The same logic applies for *Benchmark_plus* where the FC^+_{enz} is computed as:

$$\begin{aligned} FC^+_{enz} &= K^+ * FC^0_{enz} \\ &\text{where } K^+ = 1 + \Delta^+ \\ &\text{and } \Delta^+ = [t(\mu^+, \omega^+) - t(\mu^0, \omega^0)] \end{aligned} \tag{3.7}$$

Δ^+ refers to a difference between *Benchmark_plus* observed sign and weight and *benchmark_zero* observed sign and weight.

When applying these equations, the qualitative predictions presented in Table 3.4 for MajS are converted into quantitative values presented in Table 3.5.

3.3.5 MajS' integration into the metabolic network compared to Probregnet and Iggy.

As explained in Section 2.3.4, we integrated the quantified predictions of MajS into the metabolic network based on the Probregnet pipeline (see Figure 2.7).

We then compared the net ATP productions upon HIF1A perturbation obtained with MajS, Iggy and Probregnet predictions. We also applied two different mathematical paradigms, LSEI and MCMC, explained in Section 2.3.4. The results of this comparison are presented in Figure 3.7.

We concluded that similarly to Probregnet, MajS makes it possible to see the repercussion of HIF1A perturbation on net ATP production. This is not the case for Iggy. However, MajS gives stronger repercussions than Probregnet. Besides, the MCMC paradigm is more

Table 3.5 – MajS quantified predictions upon perturbations of HIF1A. For *Benchmark_zero* we applied Equation 3.5, for *Benchmark_minus* we applied Equation 3.6 and for *Benchmark_plus* we applied Equation 3.7. The colours are focused on *Benchmark_zero* and *Benchmark_plus*; the enzymes predicted with "+" sign appear in green. The ones predicted with "-" appear in pink.

	Benchmark_minus	Benchmark_zero	Benchmark_plus
ALDOA	0.6624316	0.8832421	1.1040527
ENO1	0.6384940	0.8513253	1.0641566
ENO2	0.6313452	0.8417936	1.0522421
ENO3	0.7382321	0.9843095	1.2303869
GAPDH	0.6427238	0.8569651	1.0712063
HK1	0.6920324	0.9227099	1.1533874
HK2	0.6691657	0.8922209	1.1152761
HK3	0.7576660	1.0102213	1.2627767
LDHA	0.6479622	0.8639496	1.0799370
PFKL	0.7898813	1.0531751	1.3164689
PGK1	0.9179787	1.2239717	1.5299646
SLC2A1	0.8263782	1.1018377	1.3772971
PDHA1	1.1073790	0.8859032	0.8859032
PDHA2	1.2309358	0.9847486	0.9847486
PDHB	0.9743527	0.7794822	0.7794822

sensible for the three methods than LSEI to measure a perturbation impact.

3.4 Exploring other sign-consistency rules

In parallel with this work, I co-supervised a research project on a topic closely related to MajS. Two students, Gen LI and Khaled EL GHAMMARTI from *Ecole Centrale de Nantes* conducted this project for two months and a total of 60 working hours. The project consisted in an implementation of a different sign-consistency rule (see Section 1.4.1) similar to Iggy or MajS to relate the sign of a node with its predecessors. Implementation of this method can be found on GitHub: `https://github.com/rami3l/ecn-pappl-asp.git`.

Weighted Sign-consistency approach

In this method which is sign-consistency based, we assign to each node of the graph an integer from the set $-L, ..., +L$. This integer is called the signed weight. This integer L represents the maximum integer value a node can take. Similarly to Iggy and MajS, this

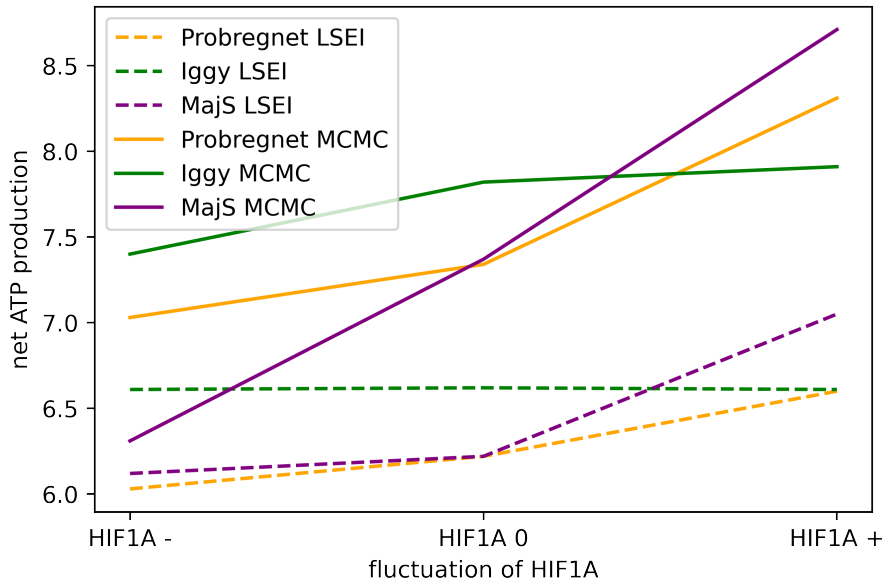


Figure 3.7 – **Impact of MajS prediction upon HIF1A perturbation over net ATP production compared to Probregnet and Iggy.** In orange, Probregnet enzymes' predictions are integrated with the metabolic network. In purple MajS enzymes' predictions are quantified and then integrated with the metabolic network, and in green, Iggy enzymes' predictions. In the dashed line, the LSEI paradigm is applied; in the solid line, the MCMC paradigm is applied.

method requires a file of partial observations of the graph nodes. Thus, each node from this file is associated with an observed integer value ranging from -L to +L.

Rules are defined according to the logical problem and are based on Iggy rules. In the following paragraph, we present some of the logical rules implemented in the ASP program:

1. Configuration: introduction of a predicate called *signedWeight* which represents an integer ranging from -L to +L.

```
signedWeight(-L..L).
```

   This line stipulates that signed weight is comprised between -L and +L.

2. Guessing: generate an associated integer value for each node between -L and L.

```
1{node_color(N, C): signedWeight(C)}1 :- node(N).
```

   Based on the Configuration rule, the predicate *node_color* of arity 2 is associating an integer value (C) with $|C| < L$ to each node of a graph (N). This ASP line

makes the solver enumerate all possible different assignments to each node of the graph.

3. Observation satisfaction: for each observed node, the model behaviour must match the biological observation data. The ASP implementation is as follows:

```
forbidden(N, C) :- observed(N, Z), signedWeight(C), C!=Z.
```

The predicate *observe* is formed with the observation file. The predicate *forbidden* represents all unauthorized associations. This line stipulates that we cannot have an observed node with an integer that goes out of the range between -L and L.

4. Justification of the change in all predecessors. Recall that an input node is a node without parent nodes. The integer value associated with a non-input node will be the mean of its predecessors' influences. The notion of influence is explained in Section 3.2.1. The ASP implementation is as follows:

```
node_color(J, C) :- node(J), not input(J), S = #sum{F
    :influence(I,J, F), F !=0},T = #count{I :influence(I, J, F), F
    !=0}, C = S/T.
```

The ASP operation "#sum" is used to retrieve the sum of the signed influences received upon node J inside the variable S. The operation "#count" is used to count, inside the variable T, the total number of J's predecessors (I) having signed influences. C computes the mean of the signed predecessors' influences upon node J. The predicate *influence* is of arity 3, and for a node, J gives its parent, I and the influence received from its parent, F. The *node_color* predicate of arity 2 associates a node and its signed weight. This line stipulates that the weighted sign C of node J equals the sum of its signed ("+"/"-") received influences over the total number of predecessors.

5. Justification of zero change: A node is signed as zero if the sum of its received influences is zero.

```
forbidden(J, 0) :- influence(J), #sum{F : influence(J, F)} != 0.
```

The predicate *influence*/1 means that node J received at least one influence. This line stipulates that it is forbidden to have a node J with an associated integer equal to 0 if the sum of received influences is different from 0.

6. Verification

```
:- forbidden(N, C), node_color(N, C).
```

This line stipulates that we cannot have a node with an integer which is forbidden.

As for MajS and Iggy, the program will make repairs by adding new influences and optimisation to have solutions with the minimum number of repairs.

Application on a toy example

The toy IG is composed of 4 nodes and 4 edges with 3 inhibitions and one activation; two nodes, *rpmC* and *rspP*, are observed respectively, at levels -2 and 2. The integer L is fixed to 2. The method proposes one optimal answer set presented in Figure 3.8.

For this solution, there is one repair over node *fnr*, which consists in adding a new negative influence to *fnr*. Now let us focus on *rpmc*; this node has two predecessors: *fnr* and *arcA*. *rpmc* is associated with the integer -2 which is consistent with the configuration with *fnr* equals to -2 and *arcA* equals to 0 because only *fnr* influence is considered (see rule entitled "justification of the change in all predecessors"). The same reasoning is applied to *rpsP* node.

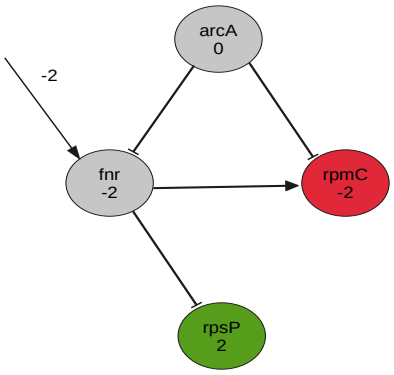


Figure 3.8 – The interaction graph comprises 4 nodes and 4 edges (3 inhibition and 1 activation). The two observed nodes are red (-) and green (+). The unobserved nodes are in grey.

This approach provides us with an interesting result on a small network. This method now requires to be tested on larger networks. Nonetheless, this method could inspire us to reduce the search space by using a signed weight instead of a sign and a weight, as done with MajS.

3.5 Discussion and Conclusion

We present in our study a new logical approach, MajS, implemented in Answer Set Programming. MajS takes as input an interaction graph and a set of discrete observations. Discrete observations are expressed in the forms of “+", “-", or “0" signs in some of the graph nodes. This information is extracted from gene expression datasets. MajS tests the consistency between the majoritarian sign of a node's direct predecessors and the node's sign; detects and repairs inconsistencies, and predicts unobserved nodes. MajS' prediction is given as a sign and a weight assigned to each unobserved node, where the weight represents the sign confidence. In addition, MajS outputs information concerning the prediction distribution across all consistent optimal answer sets or models. MajS was tested on two networks derived from the HIF-1 signalling pathway. These two networks were reduced with Alzheimer's disease (AD) Microarray and HUVECS RNA-Seq datasets. They are composed of respectively, 94 and 81 nodes and 285 and 233 edges.

For both studied networks, MajS finds results in a couple of minutes which opens perspectives to handle larger networks. MajS outputs informative predictions on all unobserved nodes such as the majoritarian sign, the average weight, and the standard deviation of this weight in all benchmarks studied.

Several results are obtained upon comparison to a similar discrete and logical approach, Iggy. First, MajS' coverage is higher than Iggy's in all our tested benchmarks (see Section 3.3.2 and 3.3.3). Second, MajS is more sensitive to the perturbation of our system thanks to the notion of weight (see Section 3.3.3). Third, MajS is more reliable when compared to *in vitro* real data (see Section 3.3.2). With respect to Iggy's implementation, MajS logical rules better constrain the network-data integration problem. Thus, MajS can propose a wider view, together with a distribution analysis, of all optimal answer sets. This is hard to accomplish with Iggy; for example, enumerating all optimal answer sets for the AD Benchmark (see Section 3.3.3) outputs 10^{10} solutions after 21 days without giving a complete solution.

We also compared MajS to a Bayesian approach, Probregnet. Focusing on the enzyme nodes prediction, MajS and Probregnet agree (11/15) on a similar dynamic regarding the evolution of enzyme predictions across HIF1A perturbation. The differences observed between MajS and Probregnet occur for enzymes for whom Probregnet prediction was not significantly varying across different perturbations (see Section 3.3.3). As with Iggy [14], MajS uses fewer input data than Probregnet. Also, Probregnet and MajS are used for

specific purposes. Probregnet allows modelling small networks; whereas MajS and Iggy are adapted for larger networks. Both can measure the impact of a single node perturbation in a system. However, MajS can model multiple nodes' perturbations. Probregnet works on a specific condition, whereas MajS deals with differential comparison between two conditions. Nevertheless, it is interesting to obtain quantitative predictions, as proposed by Probregnet, that easily adapt to linear programming metabolic modelling. MajS goes a step further than Iggy by outputting finer discretised predictions.

All in all, we have proposed MajS, a new method, fast and reliable, that tests consistency and predicts the change of expression on unobserved nodes (sign and weights) when comparing a regulatory network with a gene expression dataset. Our method is applied to perturbed data. In particular, we applied this method to test consistency between the HIF-1 signalling pathway and a HIF1A overexpressed dataset. Besides, MajS by predicting weights allows us to have a more refined prediction and proposes predictions which are sensitive to system perturbation.

A perspective of this work would focus on integrating MajS predicted enzymes into a metabolic network model and compare the results obtained concerning Probregnet's full pipeline. MajS, contrary to Probregnet, also handles better network inhibitions by relying on network topology. Indeed, Probregnet authors added a correction term in their publication to cope with inhibitions [48].

One of the possible limitations of MajS could be the repair process. Different repairs can be used in an interaction graph (e.g. remove, add or flip by changing the sign of edges; remove, add or flip nodes). In our study, we choose to add influences (positive or negative) to agree with the majority sign rules. Indeed, combining the repair process can appear to be a good idea, but it can become time-consuming in practice. However, testing other repairs process could be interesting for future work. Another limitation of MajS is that it keeps the sign of observed nodes even if they are inconsistent. In the case of many inconsistent nodes, we should question the quality of the experimental data or the interaction graph. Because of these inconsistencies, the method may give predictions which are not relevant. One idea to take care of this unreliable experiment could be to use a smaller weight to represent some observed nodes with a low confidence observation and to propagate this weight inside the IG.

General conclusion and perspectives

3.6 Conclusion

The objectives of this thesis were multiple. The main one was to predict the biological system's response to a perturbation. Different biological layers inside an organism can be impacted during a perturbation. We were particularly interested in two interrelated networks, the gene regulatory network and the metabolic network. The gene regulatory network is interesting to study during a perturbation because it affects the expression of specific genes. Besides, gene regulation is an essential element in the adaptability of an organism. In addition, the primary goal of the gene regulatory network is to generate regulated production of transcripts (mRNA, proteins). However, proteins impact another biological network essential for an organism's survival, the metabolic network, whose primary purpose is to modelise the production of energy in the form of ATP. Proteins, specifically enzymes, catalyse biochemical reactions within the metabolic network. But, this link goes in both directions because specific metabolites produced by biochemical reactions can also affect the expression of particular genes products such as transcription factors. Understanding how the regulatory and metabolic networks react to certain perturbations is crucial for treating certain diseases (such as neurodegenerative diseases, autoimmune diseases, and diabetes).

The central objective of my thesis was to sudy the different strategies to model an organism and its different biological layers. Therefore, during this thesis, we seek to model the gene regulatory and metabolic networks and their interactions. Initially, we were interested in the gene regulatory network and listed some modelling approaches for this type of network. We selected two different methods that seemed interesting to compare, a sign consistency approach, Iggy, and a Bayesian approach, Probregnet. These approaches can be used on large networks [6]. Indeed, another objective of the thesis is to model biological networks at the scale of the entire network, which can attain several thousand nodes. We also searched for an approach to model metabolic networks, and we decided to

stick with the most common method, which is a constraint-based approach called Flux Balance Analysis (FBA). Indeed, this approach allows the study of large networks (more than thousands of nodes) and can be applied to a large panel of organisms depending on prior knowledge available, which is the metabolic network. Besides, we aim to study the interaction between both models, so we turn our research on already existing approaches which allow taking into account this integration. Some of these approaches are presented in Chapter 1. We concluded that most of those approaches are used on well-known organisms and require a lot of input data, such as specific parameters or thousands of gene expressions.

Our thesis motivation comes from the lack of an approach which allows an integration of the regulatory/metabolic network without requiring too much input data and applies to a large spectrum of organisms. We are going to list some of the main achievements we obtain during this thesis work.

3.6.1 Iggy and Probregnet comparison

Iggy and Probregnet use different inputs for modelling regulatory networks

Although these two approaches make it possible to predict *in silico* the effect of perturbations on the regulatory network, they need different inputs. Both methods work with prior knowledge, such as a graph representing the regulatory network and experimental observations data. However, in terms of experimental data, Probregnet needs at least ten samples (e.g. patients or cells) and only one observed condition (healthy or perturbed). In contrast, Iggy needs two samples in two different conditions. For the regulatory network, Probregnet works on a network composed of about 100 nodes but did not scale well on a tested network of about 1000 nodes. At the same time, Iggy handles large-scale network efficiency (see [59]). Besides, Probregnet needs to represent the regulatory network as a directed acyclic graph. In contrast, Iggy uses a directed graph, so the pre-treatment is less important with Iggy on the network structure.

Iggy is comparable with Probregnet in terms of enzyme predictions on the Microarray dataset

We focus on the computational predictions of 15 enzymes which are known to have an impact on biochemical reactions of the brain metabolism. The Microarray dataset is composed of gene expression data extracted from the Hippocampus brain region of

Alzheimer's Disease (AD) patients and healthy individuals. The regulatory network is from the HIF signalling pathway as HIF1A is a key protein which is a putative therapeutic target for neurodegenerative disease.

Our results on the Microarray dataset were that Iggy and Probregnet showed very similar (73.3% of agreement) computational enzyme predictions across the same HIF1A perturbation (HIF1A under-expressed, HIF1A unchanged, HIF1A overexpressed). Among the disagreements, we count a total of 4 enzymes predicted differently between the two approaches, and only one of them has a significantly different prediction (over a fixed threshold of 0.1).

Iggy agrees with *in vitro* perturbed data on enzyme evolution in the RNA dataset

The RNA dataset is composed of 6 human umbilical vein endothelial cells (HUVECS); 3 cells are adenovirally over-expressing HIF1A protein, and the 3 others are normally expressing HIF1A. In this second dataset, we obtained different enzyme predictions (66.6% of agreement) using both modelling approaches; however, Iggy's predictions followed experimentally measured results on enzyme expression. In addition, we compared all the sign predictions for all the nodes present in the graph (81), and Iggy agrees with real data for 65% of the nodes, while Probregnet agrees only for 43.75%.

Iggy is a good candidate for modelling large regulatory networks and taking into account inhibition and complex formation

In Chapter 2, we concluded that the logical approach, Iggy, was a good candidate for modelling large regulatory networks and allowed us to predict a perturbation's effect on the expression of some genes. Besides, Iggy is able to account for different biological interactions such as activation and inhibition but also complex formation.

Nevertheless, this approach does not allow easy integration with the metabolic network due to its qualitative predictions. That is the main reason for implementing a new logical strategy using the Answer set programming Language, MajS.

3.6.2 MajS method

MajS predicts more unobserved species than Iggy inside a regulatory network

This new developed logical approach, MajS, presented in Chapter 3, is also applied to the Microarray and RNA datasets. MajS, as Iggy, needs a graph representing the regulatory network and a list of partial observations of the graph nodes with discrete values ("+","-", "0"). However, MajS is implemented differently with other logic rules than Iggy. In particular, MajS is enriched by the notion of weight associated with the sign of a node. The other notable difference is that MajS is based on the majority rule, where Iggy will check that there is at least one influence that justifies the sign of a node, and MajS checks that the majority of influences explain the sign of the node.

In this chapter, for each dataset, we tested different benchmarks. Changes inside the observation list characterize each benchmark. All benchmarks are detailed in Section 3.3.1.

For the RNA dataset, we designed two benchmarks. The first is deduced directly from observations of the gene expression data. The other benchmark is based on the first with certain nodes that have been modified to see the impact of these modifications on the predictions. For benchmark 1, MajS is able to predict all the unobserved species, while Iggy predicts only 59% of unobserved species. For benchmark 2, both predict 100 % of unobserved species.

For the Microarray dataset, we tested 3 different benchmarks. The difference between them is the HIF1A value which is either set to "+", "-", or "0". Iggy predicts only 20% of unobserved species when HIF1A = "-", 88% when HIF1A1="0" and 85% when HIF1A ="+" whereas MajS predicts all unobserved species across the three benchmarks.

In conclusion, MajS predicts more unobserved species than Iggy, with a coverage of 100% for all the studied benchmarks.

MajS is more sensitive to a system perturbation compared to Iggy

This is illustrated in the Microarray dataset. We applied MajS and Iggy on three benchmarks and collected the computational enzyme prediction. We observed that between two different perturbations of HIF1A (HIF1A set to "0" and HIF1A set to "+") the enzyme is signed the same way with Iggy, so we do not see any repercussion of HIF1A perturbation. In contrast, using MajS, we could see a repercussion for most of the enzymes on the same benchmarks thanks to the notion of weight. MajS is more sensitive to HIF1A perturbation than Iggy for this case study.

MajS predictions are closer to real data

We compared MajS and Iggy predictions with the *in vitro* perturbed data for the RNA case study. We computed a significance score and deducted MajS as a score either equal to Iggy's score or higher for most of the genes (around 60%). We deduced that MajS obtained higher confidence on the predicted sign than Iggy for this case study.

MajS propose finer-grained prediction for enzymes

MajS provides fine-grained discrete predictions by outputting the weight of the predicted sign as additional information. This weight allows, compared to Iggy, to have variability among nodes even if they are signed the same way but also to be more sensitive to system perturbation. To conclude, we have developed a logical approach that makes it possible to model the effect of a perturbation on the regulatory network and to have more easily quantifiable predictions, which was an essential step for integrating into the metabolic network.

MajS, as Probregnet and contrary to Iggy, allows measuring the impact of a perturbation on the metabolism

In Section 3.3.5, we compared the repercussion of Iggy and Probregnet enzymes' quantified predictions upon HIF1A perturbation. We concluded that MajS, contrary to Iggy, makes it possible to see HIF1A perturbation impact over net ATP production. This repercussion on the net ATP production, which is also visible with the Probregnet pipeline, agrees with the literature. Indeed, during a neurodegenerative disease, the net ATP production decreases in the Brain [63]. This is confirmed *in silico*, after performing a FBA (see Figure 2.7), the net ATP production is around 9.56 in a healthy brain, whereas in Figure 3.7 the net ATP production is of maximum 7.34 without perturbation of HIF1A. The net ATP production without HIF1A perturbation corresponds to the net ATP production in Alzheimer's disease (AD) brain because we only consider the enzyme Fold-Change between AD patients and healthy individuals. When HIF1A expression increases, we can see that the net ATP production is also increasing. In this sense, by restoring net ATP production, it could be a putative therapeutic target for neurodegenerative disease. In [55], the authors already studied the effect of HIF1A on ATP production and reached the same conclusion on the fact that an increase of HIF1A corresponds to an increase of ATP. The authors deduced that HIF1A is a putative drug target for neurodegenerative

diseases.

We are now going to list some of the perspectives we have after this thesis work.

3.7 Perspective

3.7.1 Works on future MajS integration with metabolic network

The next step will be to integrate MajS in a cleaner way than currently performed. We have identified some points which could enhance the integration process of MajS.

Firstly, **propose more generic quantification formulas** than the one presented in Section 3.3.5.

Secondly, in order to improve the quantification process and have more refined predictions, we ant to **work on weight propagation** in the logic program.

Thirdly, we want to continue the integration by constraining the fluxes of the reactions mainly via enzymes, as done in other approaches presented in Section 1.6. However, we intend to **test different types of integration**; for example, rather than taking the average of the fold changes as Probregnet (see Figure 2.7), we believe that taking the minimum if the enzymes form a complex ("AND" gate) and the average if the enzymes compete ("OR" gate) is a good alternative.

Finally, we want to work on other mathematical paradigms. MCMC is more appropriate than LSEI for biological data, which are often underdetermined because of the lack of data. However, we could also **look for different mathematical paradigms** to apply and find the most relevant ones to solve inverse linear problems.

In the end, we will **test our method on other case studies**. Biological data, where the perturbations and the repercussion on metabolic data (biomass, net ATP production or production of one or more metabolites) have also been verified *in vitro*, could allow us to refine and validate our model. A model organism, *S. Cerevisiae* possesses such data.

3.7.2 Application of our method to another organism, S. Cerevisiae

S. Cerevisiae is a model organism for which many data are already available, including gene expression data before/after disruption and effects on yeast growth. Therefore, this organism could validate our approach and the integration between the two networks we developed. We have already reconstructed a gene regulatory network using the yeastract database, which gives interactions based on literature and data. This network is around 1000 nodes. We have also retrieved the yeast metabolic network (available in the BIGG database). But before we can apply our method, we face different challenges that will be our future research projects. The first challenge is **to discretise the data optimally** to generate a correct observation file. We must also work on**scaling up our method** by optimising the ASP code and **adapting our code for this studied organism**. Once all this is done, we will be able to **compare our results with the experimental data**, and we will be able to predict the effect of new perturbations.

BIBLIOGRAPHY

[1] Rajeshwar Govindarajan, Jeyapradha Duraiyan, Karunakaran Kaliyappan, and Murugesan Palanisamy, « Microarray and its applications », *in*: *Journal of Pharmacy & Bioallied Sciences* 4.*Suppl 2* (Aug. 2012), S310–S312, ISSN: 0976-4879, DOI: 10.4103/0975-7406.100283, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3467903/.

[2] Kimberly R. Kukurba and Stephen B. Montgomery, « RNA Sequencing and Analysis », *in*: *Cold Spring Harbor protocols* 2015.*11* (Apr. 2015), pp. 951–969, ISSN: 1940-3402, DOI: 10.1101/pdb.top084970, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4863231/.

[3] Ron Edgar, Michael Domrachev, and Alex E. Lash, « Gene Expression Omnibus: NCBI gene expression and hybridization array data repository », eng, *in*: *Nucleic Acids Research* 30.*1* (Jan. 2002), pp. 207–210, ISSN: 1362-4962, DOI: 10.1093/nar/30.1.207.

[4] Hiroyuki Ogata, Susumu Goto, Kazushige Sato, Wataru Fujibuchi, Hidemasa Bono, and Minoru Kanehisa, « KEGG: Kyoto Encyclopedia of Genes and Genomes », *in*: *Nucleic Acids Research* 27.*1* (Jan. 1999), pp. 29–34, ISSN: 0305-1048, DOI: 10.1093/nar/27.1.29, URL: https://doi.org/10.1093/nar/27.1.29.

[5] Charles J. Norsigian, Neha Pusarla, John Luke McConn, James T. Yurkovich, Andreas Dräger, Bernhard O. Palsson, and Zachary King, « BiGG Models 2020: multi-strain genome-scale models and expansion across the phylogenetic tree », eng, *in*: *Nucleic Acids Research* 48.*D1* (Jan. 2020), pp. D402–D406, ISSN: 1362-4962, DOI: 10.1093/nar/gkz1054.

[6] Hanif Yaghoobi, Siyamak Haghipour, Hossein Hamzeiy, and Masoud Asadi-Khiavi, « A Review of Modeling Techniques for Genetic Regulatory Networks », *in*: *Journal of Medical Signals and Sensors* 2.*1* (2012), pp. 61–70, ISSN: 2228-7477, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3592506/.

[7] Claudio Angione, « Human Systems Biology and Metabolic Modelling: A Review—From Disease Metabolism to Precision Medicine », en, *in*: *BioMed Research International* 2019 (June 2019), Publisher: Hindawi, e8304260, ISSN: 2314-6133, DOI: 10.1155/2019/8304260, URL: https://www.hindawi.com/journals/bmri/2019/8304260/.

[8] Sven Thiele, Luca Cerone, Julio Saez-Rodriguez, Anne Siegel, Carito Guziołowski, and Steffen Klamt, « Extended notions of sign consistency to relate experimental data to signaling and regulatory network topologies », *in*: *BMC Bioinformatics* 16.*1* (Oct. 2015), p. 345, ISSN: 1471-2105, DOI: 10.1186/s12859-015-0733-7, URL: https://doi.org/10.1186/s12859-015-0733-7.

[9] Sven Thiele, Sandra Heise, Wiebke Hessenkemper, Hannes Bongartz, Melissa Fensky, Fred Schaper, and Steffen Klamt, « Designing Optimal Experiments to Discriminate Interaction Graph Models », *in*: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16.*3* (May 2019), Conference Name: IEEE/ACM Transactions on Computational Biology and Bioinformatics, pp. 925–935, ISSN: 1557-9964, DOI: 10.1109/TCBB.2018.2812184.

[10] Filipe Gouveia, Inês Lynce, and Pedro T. Monteiro, « Revision of Boolean Models of Regulatory Networks Using Stable State Observations », *in*: *Journal of Computational Biology* 27.*2* (Feb. 2020), pp. 144–155, DOI: 10.1089/cmb.2019.0289, URL: https://www.liebertpub.com/doi/10.1089/cmb.2019.0289.

[11] MARKUS W. Covert, CHRISTOPHE H. Schilling, and BERNHARD Palsson, « Regulation of Gene Expression in Flux Balance Models of Metabolism », en, *in*: *Journal of Theoretical Biology* 213.*1* (Nov. 2001), pp. 73–88, ISSN: 0022-5193, DOI: 10.1006/jtbi.2001.2405, URL: https://www.sciencedirect.com/science/article/pii/S0022519301924051.

[12] Tomer Shlomi, Yariv Eisenberg, Roded Sharan, and Eytan Ruppin, « A genome-scale computational study of the interplay between transcriptional regulation and metabolism », *in*: *Molecular Systems Biology* 3 (Apr. 2007), p. 101, ISSN: 1744-4292, DOI: 10.1038/msb4100141, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1865583/.

[13] Sriram Chandrasekaran and Nathan D. Price, « Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in Escherichia coli and Mycobacterium tuberculosis », eng, *in*: *Proceedings of the National Academy of Sciences of*

the United States of America 107.*41* (Oct. 2010), pp. 17845–17850, ISSN: 1091-6490, DOI: 10.1073/pnas.1005139107.

[14] Sophie Le Bars, Jérémie Bourdon, and Carito Guziolowski, « Comparing Probabilistic and Logic Programming Approaches to Predict the Effects of Enzymes in a Neurodegenerative Disease Model », en, *in*: *Computational Methods in Systems Biology*, ed. by Alessandro Abate, Tatjana Petrov, and Verena Wolf, Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 141–156, ISBN: 978-3-030-60327-4, DOI: 10.1007/978-3-030-60327-4_8.

[15] J. E. Murray, N. Laurieri, and R. Delgoda, « Chapter 24 - Proteins », en, *in*: *Pharmacognosy*, ed. by Simone Badal and Rupika Delgoda, Boston: Academic Press, Jan. 2017, pp. 477–494, ISBN: 978-0-12-802104-0, DOI: 10.1016/B978-0-12-802104-0.00024-X, URL: https://www.sciencedirect.com/science/article/pii/B978012802104000024X.

[16] Roger Bumgarner, « DNA microarrays: Types, Applications and their future », *in*: *Current protocols in molecular biology / edited by Frederick M. Ausubel ... [et al.]* 0 22 (Jan. 2013), Unit–22.1. ISSN: 1934-3639, DOI: 10.1002/0471142727.mb2201s101, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4011503/.

[17] R. J. Lipshutz, D. Morris, M. Chee, E. Hubbell, M. J. Kozal, N. Shah, N. Shen, R. Yang, and S. P. Fodor, « Using oligonucleotide probe arrays to access genetic diversity », eng, *in*: *BioTechniques* 19.*3* (Sept. 1995), pp. 442–447, ISSN: 0736-6205.

[18] Zhuo Wang, Samuel A. Danziger, Benjamin D. Heavner, Shuyi Ma, Jennifer J. Smith, Song Li, Thurston Herricks, Evangelos Simeonidis, Nitin S. Baliga, John D. Aitchison, and Nathan D. Price, « Combining inferred regulatory and reconstructed metabolic networks enhances phenotype prediction in yeast », en, *in*: *PLOS Computational Biology* 13.*5* (May 2017), Publisher: Public Library of Science, e1005489, ISSN: 1553-7358, DOI: 10.1371/journal.pcbi.1005489, URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005489.

[19] Jasin Hodzic, Lejla Gurbeta, Enisa Omanovic-Miklicanin, and Almir Badnjevic, « Overview of Next-generation Sequencing Platforms Used in Published Draft Plant Genomes in Light of Genotypization of Immortelle Plant (Helichrysium Arenarium) », *in*: *Medical Archives* 71.*4* (Aug. 2017), pp. 288–292, ISSN: 0350-199X, DOI: 10.5455/medarh.2017.71.288-292, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5585786/.

[20] Hans Peter Fischer, « Mathematical modeling of complex biological systems: from parts lists to understanding systems behavior », eng, *in*: *Alcohol Research & Health: The Journal of the National Institute on Alcohol Abuse and Alcoholism* 31.*1* (2008), pp. 49–59, ISSN: 1930-0573.

[21] Ana Conesa and Stephan Beck, « Making multi-omics data accessible to researchers », en, *in*: *Scientific Data* 6.*1* (Oct. 2019), Number: 1 Publisher: Nature Publishing Group, p. 251, ISSN: 2052-4463, DOI: 10.1038/s41597-019-0258-4, URL: https://www.nature.com/articles/s41597-019-0258-4.

[22] Werner Dubitzky, Olaf Wolkenhauer, Kwang-Hyun Cho, and Hiroki Yokota, eds., *Encyclopedia of Systems Biology*, en, New York, NY: Springer New York, 2013, ISBN: 978-1-4419-9862-0 978-1-4419-9863-7, DOI: 10.1007/978-1-4419-9863-7, URL: http://link.springer.com/10.1007/978-1-4419-9863-7.

[23] Heladia Salgado, Socorro Gama-Castro, Agustino Martínez-Antonio, Edgar Díaz-Peredo, Fabiola Sánchez-Solano, Martín Peralta-Gil, Delfino Garcia-Alonso, Verónica Jiménez-Jacinto, Alberto Santos-Zavaleta, César Bonavides-Martínez, and Julio Collado-Vides, « RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in Escherichia coli K-12 », eng, *in*: *Nucleic Acids Research* 32.*Database issue* (Jan. 2004), pp. D303–306, ISSN: 1362-4962, DOI: 10.1093/nar/gkh140.

[24] V. Matys, E. Fricke, R. Geffers, E. Gößling, M. Haubrock, R. Hehl, K. Hornischer, D. Karas, A. E. Kel, O. V. Kel-Margoulis, D.-U. Kloos, S. Land, B. Lewicki-Potapov, H. Michael, R. Münch, I. Reuter, S. Rotert, H. Saxel, M. Scheer, S. Thiele, and E. Wingender, « TRANSFAC®: transcriptional regulation, from patterns to profiles », *in*: *Nucleic Acids Research* 31.*1* (Jan. 2003), pp. 374–378, ISSN: 0305-1048, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC165555/.

[25] M. Kanehisa and S. Goto, « KEGG: kyoto encyclopedia of genes and genomes », eng, *in*: *Nucleic Acids Research* 28.*1* (Jan. 2000), pp. 27–30, ISSN: 0305-1048, DOI: 10.1093/nar/28.1.27.

[26] Vladimir Lifschitz, « What is Answer Set Programming? », *in*: AAAI'08 (2008), pp. 1594–1597.

[27] Guillaume Collet, Damien Eveillard, Martin Gebser, Sylvain Prigent, Torsten Schaub, Anne Siegel, and Sven Thiele, « Extending the Metabolic Network of Ectocarpus Siliculosus Using Answer Set Programming », en, *in*: *Logic Programming and Nonmonotonic Reasoning*, ed. by Pedro Cabalar and Tran Cao Son, Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 245–256, ISBN: 978-3-642-40564-8, DOI: 10.1007/978-3-642-40564-8_25.

[28] Carito Guziolowski, Santiago Videla, Federica Eduati, Sven Thiele, Thomas Cokelaer, Anne Siegel, and Julio Saez-Rodriguez, « Exhaustively characterizing feasible logic models of a signaling network using Answer Set Programming », *in*: *Bioinformatics* 29.*18* (Sept. 2013), pp. 2320–2326, ISSN: 1367-4803, DOI: 10.1093/bioinformatics/btt393, URL: https://doi.org/10.1093/bioinformatics/btt393.

[29] Rongming Liu, Liya Liang, Emily F. Freed, Alaksh Choudhury, Carrie A. Eckert, and Ryan T. Gill, « Engineering regulatory networks for complex phenotypes in E. coli », en, *in*: *Nature Communications* 11.*1* (Aug. 2020), Number: 1 Publisher: Nature Publishing Group, p. 4050, ISSN: 2041-1723, DOI: 10.1038/s41467-020-17721-4, URL: https://www.nature.com/articles/s41467-020-17721-4.

[30] S. A. Kauffman, « Metabolic stability and epigenesis in randomly constructed genetic nets », en, *in*: *Journal of Theoretical Biology* 22.*3* (Mar. 1969), pp. 437–467, ISSN: 0022-5193, DOI: 10.1016/0022-5193(69)90015-0, URL: https://www.sciencedirect.com/science/article/pii/0022519369900150.

[31] Melody K. Morris, Julio Saez-Rodriguez, Peter K. Sorger, and Douglas A. Lauffenburger, « Logic-based models for the analysis of cell signaling networks », eng, *in*: *Biochemistry* 49.*15* (Apr. 2010), pp. 3216–3224, ISSN: 1520-4995, DOI: 10.1021/bi902202q.

[32] Abhishek Garg, Alessandro Di Cara, Ioannis Xenarios, Luis Mendoza, and Giovanni De Micheli, « Synchronous versus asynchronous modeling of gene regulatory networks », *in*: *Bioinformatics* 24.*17* (Sept. 2008), pp. 1917–1925, ISSN: 1367-4803, DOI: 10.1093/bioinformatics/btn336, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2519162/.

[33] Réka Albert and Hans G. Othmer, « The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster »,

eng, *in*: *Journal of Theoretical Biology* 223.*1* (July 2003), pp. 1–18, ISSN: 0022-5193, DOI: 10.1016/s0022-5193(03)00035-3.

[34] Martin Hopfensitz, Christoph Mussel, Christian Wawra, Markus Maucher, Michael Kuhl, Heiko Neumann, and Hans A. Kestler, « Multiscale binarization of gene expression data for reconstructing Boolean networks », eng, *in*: *IEEE/ACM transactions on computational biology and bioinformatics* 9.*2* (2012), pp. 487–498, ISSN: 1557-9964, DOI: 10.1109/TCBB.2011.62.

[35] Meike Dahlhaus, Andre Burkovski, Falk Hertwig, Christoph Mussel, Ruth Volland, Matthias Fischer, Klaus-Michael Debatin, Hans A. Kestler, and Christian Beltinger, « Boolean modeling identifies Greatwall/MASTL as an important regulator in the AURKA network of neuroblastoma », eng, *in*: *Cancer Letters* 371.*1* (Feb. 2016), pp. 79–89, ISSN: 1872-7980, DOI: 10.1016/j.canlet.2015.11.025.

[36] Robert G. Cowell, « Local Propagation in Conditional Gaussian Bayesian Networks », *in*: *J. Mach. Learn. Res.* 6 (Dec. 2005), pp. 1517–1550, ISSN: 1532-4435.

[37] Charles Nicholson, Leslie Goodwin, and Corey Clark, « Variable neighborhood search for reverse engineering of gene regulatory networks », en, *in*: *Journal of Biomedical Informatics* 65 (Jan. 2017), pp. 120–131, ISSN: 1532-0464, DOI: 10.1016/j.jbi.2016.11.010, URL: https://www.sciencedirect.com/science/article/pii/S1532046416301733.

[38] Paul Dagum and Michael Luby, « Approximating probabilistic inference in Bayesian belief networks is NP-hard », en, *in*: *Artificial Intelligence* 60.*1* (Mar. 1993), pp. 141–153, ISSN: 0004-3702, DOI: 10.1016/0004-3702(93)90036-B, URL: https://www.sciencedirect.com/science/article/pii/000437029390036B.

[39] Jing Yu, V Anne Smith, Paul P Wang, Alexander J Hartemink, and Erich D Jarvis, « Using Bayesian Network Inference Algorithms to Recover Molecular Genetic Regulatory Networks », en, *in*: *International Conference on Systems Biology (ICSB02)* (Jan. 2002), p. 10.

[40] Wei-Po Lee and Wen-Shyong Tzou, « Computational methods for discovering gene networks from expression data », eng, *in*: *Briefings in Bioinformatics* 10.*4* (July 2009), pp. 408–423, ISSN: 1477-4054, DOI: 10.1093/bib/bbp028.

[41] Juliane Schäfer and Korbinian Strimmer, « An empirical Bayes approach to inferring large-scale gene association networks », eng, *in*: *Bioinformatics (Oxford, England)* 21.*6* (Mar. 2005), pp. 754–764, ISSN: 1367-4803, DOI: 10.1093/bioinformatics/bti062.

[42] Katia Basso, Adam A. Margolin, Gustavo Stolovitzky, Ulf Klein, Riccardo Dalla-Favera, and Andrea Califano, « Reverse engineering of regulatory networks in human B cells », eng, *in*: *Nature Genetics* 37.*4* (Apr. 2005), pp. 382–390, ISSN: 1061-4036, DOI: 10.1038/ng1532.

[43] Juan I. Castrillo and Stephen G. Oliver, « Yeast systems biology: the challenge of eukaryotic complexity », eng, *in*: *Methods in Molecular Biology (Clifton, N.J.)* 759 (2011), pp. 3–28, ISSN: 1940-6029, DOI: 10.1007/978-1-61779-173-4_1.

[44] Jeffrey D. Orth, Ines Thiele, and Bernhard Ø. Palsson, « What is flux balance analysis? », *in*: *Nature biotechnology* 28.*3* (Mar. 2010), pp. 245–248, ISSN: 1087-0156, DOI: 10.1038/nbt.1614, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3108565/.

[45] Mohammadreza Yasemi and Mario Jolicoeur, « Modelling Cell Metabolism: A Review on Constraint-Based Steady-State and Kinetic Approaches », en, *in*: *Processes* 9.*2* (Feb. 2021), p. 322, ISSN: 2227-9717, DOI: 10.3390/pr9020322, URL: https://www.mdpi.com/2227-9717/9/2/322.

[46] J. S. Edwards and B. O. Palsson, « The Escherichia coli MG1655 in silico metabolic genotype: Its definition, characteristics, and capabilities », *in*: *Proceedings of the National Academy of Sciences of the United States of America* 97.*10* (May 2000), pp. 5528–5533, ISSN: 0027-8424, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC25862/.

[47] Steinn Gudmundsson and Ines Thiele, « Computationally efficient flux variability analysis », *in*: *BMC Bioinformatics* 11.*1* (Sept. 2010), p. 489, ISSN: 1471-2105, DOI: 10.1186/1471-2105-11-489, URL: https://doi.org/10.1186/1471-2105-11-489.

[48] Han Yu and Rachael Hageman Blair, « Integration of probabilistic regulatory networks into constraint-based models of metabolism with applications to Alzheimer's disease », *in*: *BMC Bioinformatics* 20.*1* (July 2019), p. 386, ISSN: 1471-2105, DOI: 10.1186/s12859-019-2872-8, URL: https://doi.org/10.1186/s12859-019-2872-8.

[49] Han Yu, Janhavi Moharil, and Rachael Hageman Blair, « BayesNetBP: An R Package for Probabilistic Reasoning in Bayesian Networks », en, *in*: *Journal of Statistical Software* 94 (June 2020), pp. 1–31, ISSN: 1548-7660, DOI: 10.18637/jss.v094.i03, URL: https://doi.org/10.18637/jss.v094.i03.

[50] Winnie S. Liang, Travis Dunckley, Thomas G. Beach, Andrew Grover, Diego Mastroeni, Douglas G. Walker, Richard J. Caselli, Walter A. Kukull, Daniel McKeel, John C. Morris, Christine Hulette, Donald Schmechel, Gene E. Alexander, Eric M. Reiman, Joseph Rogers, and Dietrich A. Stephan, « Gene expression profiles in anatomically and functionally distinct regions of the normal aged human brain », *in*: *Physiological genomics* 28.*3* (Feb. 2007), pp. 311–322, ISSN: 1094-8341, DOI: 10.1152/physiolgenomics.00208.2006, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2259385/.

[51] Anand K. Gavai, Farahaniza Supandi, Hannes Hettling, Paul Murrell, Jack A. M. Leunissen, and Johannes H. G. M. van Beek, « Using bioconductor package BiGGR for metabolic flux estimation based on gene expression changes in brain », eng, *in*: *PloS One* 10.*3* (2015), e0119016, ISSN: 1932-6203, DOI: 10.1371/journal.pone.0119016.

[52] Olivia Wilkins, Christoph Hafemeister, Anne Plessis, Meisha-Marika Holloway-Phillips, Gina M. Pham, Adrienne B. Nicotra, Glenn B. Gregorio, S. V. Krishna Jagadish, Endang M. Septiningsih, Richard Bonneau, and Michael Purugganan, « EGRINs (Environmental Gene Regulatory Influence Networks) in Rice That Function in the Response to Water Deficit, High Temperature, and Agricultural Environments », eng, *in*: *The Plant Cell* 28.*10* (Oct. 2016), pp. 2365–2384, ISSN: 1532-298X, DOI: 10.1105/tpc.16.00158.

[53] Nicholas L. Downes, Nihay Laham-Karam, Minna U. Kaikkonen, and Seppo Ylä-Herttuala, « Differential but Complementary HIF1α and HIF2α Transcriptional Regulation », eng, *in*: *Molecular Therapy: The Journal of the American Society of Gene Therapy* 26.*7* (July 2018), pp. 1735–1745, ISSN: 1525-0024, DOI: 10.1016/j.ymthe.2018.05.004.

[54] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth, « edgeR: a Bioconductor package for differential expression analysis of digital gene expression data », eng, *in*: *Bioinformatics (Oxford, England)* 26.*1* (Jan. 2010), pp. 139–140, ISSN: 1367-4811, DOI: 10.1093/bioinformatics/btp616.

[55] Ziyan Zhang, Jingqi Yan, Yanzhong Chang, Shirley ShiDu Yan, and Honglian Shi, « Hypoxia Inducible Factor-1 as a Target for Neurodegenerative Diseases », *in*: *Current medicinal chemistry* 18.*28* (Oct. 2011), pp. 4335–4343, ISSN: 0929-8673, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3213300/.

[56] Dorit Dor and Michael Tarsi, « A simple algorithm to construct a consistent extension of a partially oriented graph », en, *in*: (1992), p. 4.

[57] Yunshun Chen, Aaron T. L. Lun, and Gordon K. Smyth, *From reads to genes to pathways: differential expression analysis of RNA-Seq experiments using Rsubread and the edgeR quasi-likelihood pipeline*, en, tech. rep. 5:1438, Type: article, F1000Research, Aug. 2016, DOI: 10.12688/f1000research.8987.2, URL: https://f1000research.com/articles/5-1438.

[58] Mingzhou(Joe) Song, Chris K Lewis, Eric R Lance, Elissa J Chesler, Roumyana Kirova Yordanova, Michael A Langston, Kerrie H Lodowski, and Susan E Bergeson, « Reconstructing Generalized Logical Networks of Transcriptional Regulation in Mouse Brain from Temporal Gene Expression Data », *in*: *EURASIP Journal on Bioinformatics and Systems Biology* 2009.*1* (Jan. 2009), p. 545176, ISSN: 1687-4145, DOI: 10.1155/2009/545176, URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3171431/.

[59] Maxime Folschette, Vincent Legagneux, Arnaud Poret, Lokmane Chebouba, Carito Guziolowski, and Nathalie Théret, « A pipeline to create predictive functional networks: application to the tumor progression of hepatocellular carcinoma », *in*: *BMC Bioinformatics* 21.*1* (Jan. 2020), p. 18, ISSN: 1471-2105, DOI: 10.1186/s12859-019-3316-1, URL: https://doi.org/10.1186/s12859-019-3316-1.

[60] Martin Gebser, Benjamin Kaufmann, André Neumann, and Torsten Schaub, « clasp: A Conflict-Driven Answer Set Solver », en, *in*: *Logic Programming and Nonmonotonic Reasoning*, ed. by Chitta Baral, Gerhard Brewka, and John Schlipf, Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2007, pp. 260–265, ISBN: 978-3-540-72200-7, DOI: 10.1007/978-3-540-72200-7_23.

[61] Wei Pan, « A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments », *in*: *Bioinformatics (Oxford, England)* 18 (4 2002), pp. 546–554, ISSN: 1367-4803, DOI: 10.1093/BIOINFORMATICS/18.4.546, URL: https://pubmed.ncbi.nlm.nih.gov/12016052/.

[62] Wentao Yang, Philip Rosenstiel, and Hinrich Schulenburg, « aFold – using polynomial uncertainty modelling for differential gene expression estimation from RNA sequencing data », *in*: *BMC Genomics* 20.*1* (May 2019), p. 364, ISSN: 1471-2164, DOI: 10.1186/s12864-019-5686-1, URL: https://doi.org/10.1186/s12864-019-5686-1.

[63] Janusz Wiesław Błaszczyk, « Energy Metabolism Decline in the Aging Brain-Pathogenesis of Neurodegenerative Disorders », eng, *in*: *Metabolites* 10.*11* (Nov. 2020), E450, ISSN: 2218-1989, DOI: 10.3390/metabo10110450.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Nantes Université

Titre : Modélisation hybride, logique et linéaire pour prédire *in silico* l'effet des perturbations sur le métabolisme

Mot clés : Modélisation, Answer Set Programming, Integration de réseaux

Résumé : Les perturbations induites par une maladie, un traitement ou encore un stress environnemental affectent les organismes vivants de diverses manières. Ainsi, l'expression de certains gènes sera impactée, ce qui se répercutera sur ses produits (protéines, ARNm). Ces perturbations se propagent également via les interactions que peuvent avoir les gènes les uns avec les autres. L'ensemble de ces interactions forme le réseau de régulation, un objet important dans ces études. D'autre part, certaines protéines, appelées enzymes, ont un rôle de catalyseur des réactions biochimiques qui ont lieu au sein des organismes. L'ensemble des réactions biochimiques forme le réseau métabolique, un second objet important. Ainsi, une perturbation va impacter le réseau de régulation mais aussi le réseau métabolique puisqu'ils sont interconnectés, via les enzymes notamment. L'objectif principal de ma thèse est d'étudier l'impact d'une perturbation sur un organisme en intégrant le réseau de régulation au réseau métabolique. J'ai apporté deux contributions dans ce sens. La première est une comparaison d'une approche logique à une approche bayésienne pour savoir quelle stratégie de modélisation est la plus adaptée pour étudier les impacts des perturbations sur de grands réseaux de régulations. J'en ai déduit que bien qu'elle soit une bonne candidate, l'approche logique présente des limites de par ses prédictions qualitatives en matière d'intégration. La seconde contribution découle de ces limites, j'ai développé une méthode originale basée sur l'Answer Set Programming, MajS, proposant une prédiction plus fine de l'effet d'une perturbation sur le réseau de régulation. Ce travail ouvre la porte à de nombreuses perspectives comme une meilleure intégration des effets des perturbations au niveau du réseau métabolique et une application à d'autres organismes d'étude.

Title: Hybrid, logical and linear modeling to predict in silico the effect of perturbations on metabolism

Keywords: Computer modelling, Answer Set Programming, Networks integration

Abstract: perturbations induced by disease, treatment or environmental stress affect living organisms in various ways. Thus, the expression of certain genes will be impacted, affecting its products (proteins, mRNA). These perturbations are also propagated via the interactions that genes can have. These interactions form the regulatory network, an essential object in these studies. On the other hand, specific proteins, called enzymes, act as catalysts for the biochemical reactions occurring within organisms. All the biochemical reactions form the metabolic network, a second important object. Thus, a perturbation will impact the regulatory and metabolic networks since they are interconnected, via enzymes in particular. My thesis's main objective is to study a perturbation's impact on an organism by integrating the regulatory network into the metabolic network. I have made two contributions in this direction. The first compares a logical approach to a Bayesian approach to determine which modelling strategy is the most suitable for studying the impacts of perturbations on large regulatory networks. Although it is a good candidate, I deduced that the logical approach has limitations with its qualitative predictions regarding integration. The second contribution stems from these limits; I have developed an original method based on Answer Set Programming, MajS, offering a more refined prediction of the effect of a perturbation on the regulation network. This work opens the door to many perspectives, such as better integration of the effects of perturbations at the metabolic network level and an application to other organisms of study.