Analysis of Large-Scale Biological Networks with Constraint-Based Approaches over Static Models

Carito Guziolowski

N° d'ordre: 4014

THÈSE

présentée

devant l'Université de Rennes 1

pour obtenir

le grade de : Docteur de l'Université de Rennes 1
Mention Informatique

par

Carito Guziolowski

Équipe d'accueil : Symbiose - IRISA
École Doctorale : Matisse
Composante universitaire : IFSIC

Titre de la thèse :

Analysis of Large-Scale Biological Networks with Constraint-Based Approaches over Static Models

À soutenir le 8 janvier 2010 devant la commission d'examen

M. :	François	FAGES	Président
MM. :	Marie-France	SAGOT	Rapporteurs
	Benno	SCHWIKOWSKI	
MM. :	François	FAGES	Examinateurs
	Nicolas	LE NOVERE	
	Jacques	NICOLAS	
	Anne	Siegel	

Acknowledgements

The research I present in this thesis could not have been accomplished without interacting with all these people.

First, my profound thanks to Anne Siegel, my thesis director, for being so involved with this work.

In addition I would like to express my immense gratitude to the Symbiose project in Rennes leaded by Jacques Nicolas. In particular, thanks to Michel Le Borgne, Philippe Veber, Ovidiu Radulescu, and Sylvain Blachon, for all the discussions made around the difficult field of modeling biological data. Also in Rennes, thanks to the people that were working in the GenOuest platform, in particular to Annabel Bourdé and Anthony Bretadeau for their patience and persistence while developing the bioinformatics tools.

Abroad, I would like to thank the informatics team of Torsten Schaub in Potsdam, Germany and the bio-mathematics group of Alejandro Maass in Santiago, Chile, for their availability and interest in this research. Also, to the Chilean program "CONICYT-Ambassade de France" that financially supported my thesis work during all the past three years.

Thanks to my lovely polish family: Robert, for his continuous support in my work and for making me so nice company in France, thousands of kilometers away from my country. Thanks also to my peruvian family: to my mother, father, and brother who have always encouraged me to continue my PhD work. I cannot avoid to thank a very little person: my *petit bout* (so nicely called in French), for motivating me to take this last thesis witting period with all the calm and patience I may have.

I would also like to express my appreciation to François Fages who honored me by presiding my thesis jury. I also thank Marie-France Sagot and Benno Schwikowski for accepting the charge of being my thesis *rapporteurs*. Finally, I would like to thank Nicolas Le Novère and Jacques Nicolas that accepted to judge this work.

Contents

Introduction en Français

0.1 Biologie Systémique

Le domaine de la biologie systémique a pris son essor ces dix dernières années. Le principal but de la biologie systémique est de comprendre l'ensemble des contributions de chaque composant biologique d'un système, de manière à pouvoir expliquer une observation expérimentale concernant un processus biologique, une population de cellules, ou un organisme. Ce domaine innovant a besoin d'intégrer des données biologiques à toutes les niveaux moléculaires, ainsi que la collaboration entre chercheurs de différentes disciplines. Dans les paragraphes suivants, on décrira les méthodes et questions qui apparaissaient lorsqu'on analyse un système biologique, et les solutions existantes pour les résoudre. Notre objectif est de positioner nos recherches de manière à pouvoir postérieurement justifier leur intérêt.

0.1.1 Contexte biologique

L'expression des gènes désigne le processus dans lequel une séquence d'ADN est transformée en un composant structurel et fonctionnel de la cellule: une protéine. Ce processus a deux parties: transcription, lorsque l'ADN est transformé en ARNm, et traduction, lorsque l'ARNm est transformé en protéine (voir Fig. 1). La transcription est réalisée essentiellement par la molécule ARN-polymerase, et la traduction est effectuée par les ribosomes. Il y a quinze ans, les techniques de haut débit ont été introduites en vue de mesurer simultanément l'expression de milliers de molécules, en particulier la concentration en ARNm de milliers de gènes [LW00]. Cette étape a motivé l'apparition du champ de la biologie systémique, ayant comme principal défi de proposer des méthodes et concepts pour exploiter ces observations [Kit02, B06].

Les produits des gènes (protéines) ont différents rôles fonctionnels dans la cellule et sont exprimés sous l'effet de différents processus de contrôle dans la cellule. La décision concernant les gènes qui doivent être allumés (*on*) ou éteints (*off*) est executée par des facteurs de transcription (FT). Les FTs utilisent des métabolites/signaux comme données d'entrée de l'état actuel de l'environnement et produisent une réponse transcriptionnelle en sortie [MAJSCV06]. La régulation de l'expression génétique contrôle ainsi sur la structure et la fonction des cellules. Elle est la base de la différenciation cellulaire, ainsi que de la polyvalence et de l'adaptabilité de tout organisme. L'expression génétique peut être régulée à plusieurs niveaux: initiation de la transcription (*e.g.* par

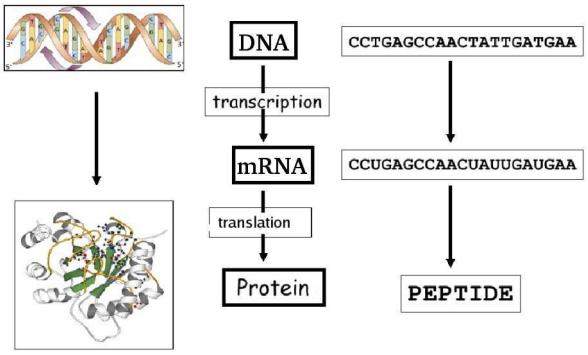


Figure 1: Etapes de l'expression génétique. Extrait du cours "Structure 3D de Protéines" du Master 2 en Bioinformatique à l'Université de Rennes 1 en 2006.

des protéines qui sont des inhibiteurs ou activateurs), terminaison prématurée de la transcription, initiation de la traduction, et post-traductionnellement. Le défi dans l'analyse de la régulation de l'expression génétique est d'étudier comment tous les éléments du processus de régulation interagissent afin de réaliser des fonctions biologiques complexes, ce qui a un impact énorme dans la médecine et la pharmacologie.

Un des types des réseaux que nous étudierons dans cette thèse est principalement lié à la *régulation de l'initiation de la transcription*. Pendant ce processus, une molécule d'ARN-polymérase s'attache à la région promotrice de la séquence ADN (gènes) et commence la transcription au long du brin d'ADN. La transcription de l'ARN-polymérase peut être régulée par des FTs qui sont des protéines ou des complexes-protéiques. Ils peuvent être des activateurs, qui

renforcent l'interaction entre l'ARN polymérase et un promoteur particulier en encourageant l'expression du gène ou des inhibiteurs, qui se lient à des séquences non codantes sur le brin d'ADN, entravant les progrès de l'ARN polymérase, et empêchant ainsi l'expression du gène. Ce type de processus de régulation peut donc être représenté par un *réseau de régulations transcriptionelles*, RRT (voir Fig. 2).

Le processus de transcription peut aussi être régulé par d'autres facteurs tels que les signaux et/ou des métabolites. Après la transcription, le processus de traduction peut aussi être régulé par d'autress types de phénomènes tels que la phosphorylation ou la séquestration. Lorsque cette information est connue, nous pouvons construire des *réseaux de signalisation*.

Les réseaux de régulation, représentés par un graphe d'interactions, peuvent contenir des informations à différents niveaux moléculaires. Nous avons mentionné deux types de réseaux: de transcription et de signalisation. Ces réseaux peuvent être de tailles différentes en fonction de la connaissance du système ou de la fonction biologique étudiée. Les travaux de cette thèse sont centrés sur les RRTs à *grande-échelle*, qui ont un graphe d'interactions composé de milliers des noeuds et ont une structure hiérarchique, et sur les réseaux de signalisation à *moyenne-échelle*, qui sont composés de centaines de

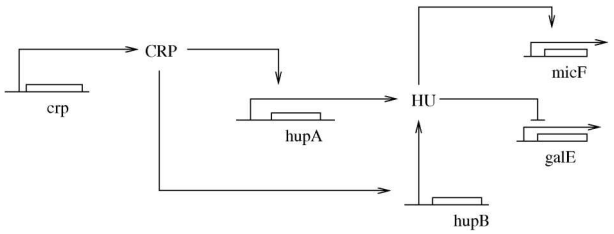


Figure 2: Une partie du réseau de régulations transcriptionelles des gènes et protéines du *E.coli*. Les noms en majuscules correspondent aux FTs (protéines): *HU* et *CRP*, qui activent ou inhibent la transcription d'autres gènes. Les flèches qui finissent par "−>" ou "−|" veulent dire que le produit de la source active ou, respectivement, inhibe la production du produit d'arrivée.

noeuds et présentent une forte connectivité. Les réseaux de régulation à *petite-échelle*, composés par des dizaines des noeuds, sont hors du cadre d'application de ces travaux.

0.1.2 Analyse intégré des réseaux de régulation

Comme mentionné précédement, une manière de représenter les interactions dans un système biologique consiste à utiliser des réseaux de régulations transcriptionnelles. Ces réseaux sont constitués d'interactions entre les facteurs de transcription et un ensemble des gènes cibles contrôlés [TB04]. Les FTs contiennent des domaines de liaison pour l'ADN, qui reconnaissent les séquences opérateurs des gènes cibles controlés [PS92]; on les appellera motifs de liaison de FTs (MLFTs). Dans le cas d'organismes microbiens il existe cinq plates-formes, qui stockent l'information concernant les MLFTs prédits et validés expérimentalement [BTR09]. Ils s'agit de RegulonDB [GCJJPG+08], MtbRegList [JGC+05], PRODORIC [MHB+03], DBTBS [SMdHN08], et CoryneRegNet [BWR+07]. Ces plates-formes contiennent des informations intéressantes concernant la construction à grande-échelle de réseaux transcriptionnels. D'après les informations qu'ils contiennent, plusieurs outils et analyses ont été proposés. Certains d'entre eux sont inclus en tant que fonctionnalités dans les plates-formes. Nous allons les classifier en trois catégories:

1. **Information sur l'organisation des réseaux:** cette catégorie inclut le stockage, par des bases des données, de toute l'information sur la régulation. On intègre aussi dans cette catégorie des outils qui fournissent une interface Web, des capacités de navigation, et d'applications Web pour naviguer dans l'information génomique. Les logiciels pour la visualisation des réseaux et l'échange d'informations sont également inclus dans cette catégorie.

2. **Analyse de séquences pour découvrir des régulations:** cette catégorie inclut des outils et des méthodologies pour l'analyse des séquences afin de mieux représenter les connaissances de régulation. Nous incluons également les analyses topologiques des réseaux de régulation. Les méthodes dans cette catégorie

effectuent les tâches suivantes:

- Prédiction des réseaux de régulation par des outils de prédiction de MLFTs, séquence promotrices, et FTs.
- Prédiction du rôle d'un FT par l'exploration des sites de l'origine de la transcription.
- Prédiction des opérons.
- Prédiction de gènes ou protéines homologues.
- Etudes topologiques de l'organisation des réseaux de régulation. Ces études ont développé une série de mesures statistiques de manière à élucider, par exemple, la hiérarchie ou des notions de connectivité d'un système [FGAPTQCV08b]. La plupart du temps, ces études n'ont besoin ni de données d'expression, ni de notions sur le rôle des régulations (activations, inhibitions).

3. **Analyse des réseaux de régulation et/ou puces à ADN** nous incluons ici des méthodes consacrées à l'analyse d'un réseau de régulation en utilisant des connaissances biologiques, puces à ADN, ou tout autre type de méthodes à haut débit. Cette catégorie de méthodes est vaste, nous pouvons, cependant, la subdiviser par le type de résultats obtenus en:

   (A) Méthodes qui formalisent les interactions du réseau en utilisant des modèles mathématiques, et en opérant des simulation ou analyses de trajectoires.

   (B) Méthodes qui intègrent des données à haut débit dans le réseaux de régulation.

   (C) Méthodes qui permettent de classer des données à haut débit et de l'information sur des MLFTs. Cette classification peut se conclure par la génération des nouvelles connaissances liées à la topologie du réseau [EBK+08]. En outre, elle peut conduire à une meilleure analyse des résultats à haut débit incertains et fournir une meilleure interprétation biologique de ces résultats [RZD+07].

   D'une manière générale, les méthodes de (A) et (B) sont essentiellement centrées sur le réseau tandis que les méthodes en (C) se concentrent sur les données. Les méthodes dans (A) sont capables d'effectuer des simulations précises de la dynamique d'un réseau de régulation réel, dans la plupart des cas d'une réseau de petite échelle. Elles ne sont pas en mesure, cependant, d'utiliser la vaste gamme des données post-génomiques. D'autre part, les méthodes en (B) relâchent la stricte formalisation du modèle de régulation pour intégrer des données génomiques à large échelle.

Les outils et méthodes bioinformatiques sont essentiels à la conception d'une connaissance biologique intégrée à partir de la grande quantité des données mises à disposition après l'ère de la génomique. On peut classer celles liées à l'étude des réseaux de régulation dans les trois catégories précisées. Les recherches que nous présentons dans cette

thèse concernent les méthodes de la catégorie 3. Les méthodes des catégories 2 et 3 peuvent générer des hypothèses à valider dans des études expérimentales. En outre, les résultats des approches des catégories 2 et 3 sont connectés. Par exemple, les méthodes qui étudient des motifs de régulation dans les réseaux (catégorie 2) sont liées à l'étude de la dynamique du système (catégorie 3).

Sur les cinq plates-formes des réseaux transcriptionnelles microbiennes à grande échelle, RegulonDB est celle qui contient le plus grand réseau de régulation lié a un organisme vivant. De ce fait, de nombreuses ressources en bioinformatique sont liées à cette plate-forme [CVSM+09]. Toutefois, une seule des cinq plates-formes (CoryneRegNet) comprend un outil qui automatise l'analyse des réseaux à grande échelle et des données issues des puces à ADN [BA08]. C'est le premier indicateur de la difficulté de cette tâche pour des réseaux à grande échelle. En fait, la difficulté des analyses reposent essentiellement sur deux points : *(i)* la difficulté du processus de reconstruction des réseaux, et *(ii)* la difficulté du raisonnement automatique sur un modèle de régulations à grande échelle.

0.2 Modélisation qualitative large-échelle

Parmi les méthodes qui raisonnent sur des réseaux à large-échelle, une partie s'appuie sur une décomposition de la structure du réseau en motifs de régulation [GRRL+03, HLPP06]. D'autres utilisent le formalisme des réseaux booléens [Kau93], par exemple [GRRL+03, CKR+04], pour calculer une mesure de consistance des régulations du réseau en accord avec les résultats expérimentaux (puces à ADN). Ils peuvent aussi générer des prédictions *in-silico* sur la variation qualitative de l'expression de certains gènes du réseau. Le principal intérêt de ces approches est qu'elles proposent une direction pour l'analyse des réseaux de régulation à grande-échelle. Toutefois, l'analyse de cohérence que ces méthodes ont proposé n'est ni automatique ni globale. La génération des prédictions peut être automatisée, cependant, elle dépendra de l'acquisition des connaissances quantitatives très précises sur les métabolites du système.

Les études qui ont proposé des méthodes pour l'analyse des réseaux à grande-échelle, utilisent des données d'expression à l'échelle d'un génome complet. Les approches qui utilisent un formalisme booléen raisonnent avec règles logiques sur les variations des molécules du réseau suite à un stress expérimental ou une perturbation génétique. L'idée principale qui sous-entend ces approches a été de mettre en place des contraintes sur les variations entre les différents états. Ce point de vue fournit une nouvelle application des approches booléennes [Kau93] dans un contexte considérablement modifié qui garantit leur validité : les variations au cours du changement d'tats plutôt que la simulation dynamique.

Sur la base de cette idée, dans [SRB+06] une méthode formelle a été proposée afin d'étudier la consistance globale d'un réseau de régulation par rapport à un jeu des données expérimentales (à grande échelle) sur la variation de l'expression génétique.

Une règle causale d'interaction générique a été utiliséé pour résoudre la question de consistance.

Les travaux dans cette thèse portent sur l'application biologique et extensions informatiques et mathématiques de cette approche : d'abord, pour tester sa validité sur un réseau transcriptionel bactérien à grande échelle, ensuite, pour appliquer cette approche aux réseaux eukaryotes complexes de transduction des signaux. Pendant ces travaux, l'approche initiale a été adaptée pour traiter des interactions complexes de régulation, surtout lorsqu'on traite des données eukaryotes. Enfin l'approche initiale a été étendue afin de pouvoir traiter des données d'expression relatives aux gènes mutés ou sur-exprimés. Trois types de sorties ont été vérifiées en permanence, adaptées, et améliorées dans toutes nos études:

- Consistance globale du réseau de régulation avec des jeux des données d'expression.
- Diagnostic des modules inconsistants.
- Prédictions sur la variation qualitative de l'expression de certaines molécules du réseau.

Ces sorties ont était générées après l'exécution d'algorithmes performants, qui peuvent traiter des grandes systèmes des contraintes et trouver une solution globale du système. Une bibliothèque Python, Bioquali, a était utilisée pour l'implémentation des programmes qui procèdent à l'analyse de consistance. De plus, des outils bioinformatiques tels qu'un site Web, un plugin Cytoscape, et un service Web, ont été développés pendant cette thèse et rendus publics.

Nos travaux viennent compléter les approches précédémment mentionnées sur les réseaux à grande échelle car notre approche est automatique et globale, et propose finalement un outil automatique pour la correction des réseaux inconsistants. Ce faisant, nous sommes en mesure de mettre en évidence les corrections minimales qui doivent être faites sur un réseau et/ou des données afin de les concilier. Contrairement aux autres méthodes qui étudient des réseaux à grande échelle, nous ne donnons pas la priorité ni aux régulations ni aux données, on considère que les deux sont sujettes à des erreurs. En comparaison avec les résultats obtenus dans [CKR$^+$04], nos prédictions, sans tenir compte de la connaissance du métabolisme du système, s'avèrent très précises (90%) après la récupération automatique des inconsistances d'un réseau à grande-échelle avec un jeu indépendant des données expérimentales.

Dans l'ensemble, nos résultats ont répondu à des questions biologiques pertinentes tels que:

- Tester la consistance globale de réseaux de régulation complexes et à grande échelle avec des données d'expression issues de puces à ADN.
- Générer manuellement et automatiquement des hypothèses pour résoudre les inconsistances entre le réseau et données d'expression.

- Prédire le rôle de régulation de facteurs de transcription comme activateurs ou inhibiteurs.
- Raisonner globalement et de manière étendue sur les états *on/off* et sur les voies qui connectent deux sommets intéressants dans un réseau de signalisation.

Etant donné le domaine de cette thèse multi-disciplinaire, les solutions et méthodes proposées répondent à différents niveaux des aspects méthodologiques, bioinformatiques, et biologiques.

Chapter 1

Introduction

1.1 Systems Biology

The systems biology field is a recently established research domain where biological systems are considered as a whole. Its main interest is to understand the joint contributions of each biological entity in order to explain an experimentally observed behavior of a cell population or an organism. This challenging domain requires the integration of biological data at all molecular levels, as well as the collaboration among multidisciplinary scientists. In the following paragraphs we will describe the methods and questions that appear when analyzing a biological system and the current solutions that exist. Our objective is to place our research in order to justify its interest.

1.1.1 Biological context

Gene expression is the process in which a DNA sequence is converted into the structures and functions of a cell: proteins. This process has two steps: transcription, when DNA is converted in mRNA, and translation, when mRNA is converted into a protein (see Fig. 1.1). Transcription is carried out mainly by the molecule RNA-polymerase, while translation is carried out by the Ribosome protein. Fifteen years ago, high-throughput techniques were introduced in order to measure simultaneously thousands of molecules, in particular the mRNA concentration of thousands of genes [LW00]. This step motivated the apparition of the systems biology field, having as a main challenge to propose methods and concepts to exploit these observations [Kit02, B06].

Gene products (proteins) have different functional roles and are expressed under different stresses. The decision about which genes should be turned *on* or *off* is executed by transcription factors (TFs). The TFs use metabolites/signals as an input information from the environmental state and give a transcriptional response as an output [MAJSCV06]. Regulation of gene expression gives a cell the control over its structure and function. It is the basis for cellular differentiation, as well as for versatility and adaptability of any organism. Gene expression can be regulated at several levels: initiation of transcription (*e.g.* by repressor or activator proteins), premature termination of transcription, initiation of translation, and post-translational. The challenge in the

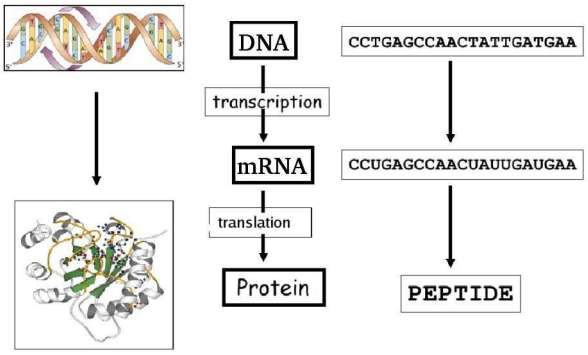


Figure 1.1: Gene expression steps. Extracted from the subject "Structure 3D de Protéines" of the Master in Bioinformatics of the University of Rennes 1 in 2006.

analysis of gene expression regulation is to study how all the components of the regulatory process interact in order to perform complex biological functions; this has an enormous impact on medicine and pharmacology.

One of the types of networks we study in this thesis is mainly related to the *regulation of the transcription initiation*. During this process a molecule RNA-polymerase attaches to the promoter region of the DNA sequence (gene) and begins transcription along all the DNA strand. The RNA-polymerase transcription can be regulated by TFs that are proteins or proteins-complexes. They can be activators, which enhance the interaction between RNA polymerase and a particular promoter encouraging the expression of the gene, or repressors, which bind to non-coding sequences on the DNA strand impeding the progress of RNA polymerase along the strand, thus, impeding the expression of the gene. This type of regulatory process can therefore be represented by a *transcriptional regulatory network* (TRN), as shown in Fig. 1.2.

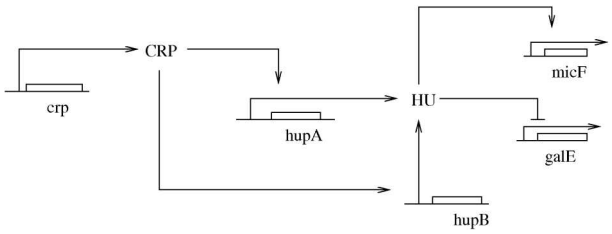


Figure 1.2: Extract of the transcriptional network of genes and proteins in *E.coli*. The names in capital letters correspond to TFs (proteins): *HU* and *CRP*, that can activate or repress other genes transcription. Arrows ending with "−>" or "−|" imply that the initial product activates or, respectively, represses production of the product of arrival.

The transcription process can also be regulated by other factors such as signals and/or metabolites. After transcription, the translation process can also be regulated

by other type of phenomena such as phosphorylation, bindings, or sequestration. When this information is known we can build *signaling networks*.

Regulatory networks, represented by a graph of interactions, may carry information at different molecular levels. We mentioned two types of them: transcriptional and signaling networks. These networks are of different sizes, depending on the knowledge of the system or on the biological function under study. The research in this thesis is centered on the *large-scale* TRNs, which graphs are composed of thousands of nodes and have a hierarchical control structure, and on the *middle-scale* signaling networks, which are composed of hundreds of nodes and hold a high connectivity between them. *Small-scale* regulatory networks (composed of tens of nodes), though well studied by different approaches, are out of the scope of this research.

1.1.2 Integrated analysis of regulatory networks

As mentioned previously, one way of representing the interactions in a biological system is by using transcriptional regulatory networks. These networks are composed of interactions among transcription factors and a set of controlled target genes [TB04]. The TFs contain DNA-binding domains that recognize the operator sequences of controlled target genes [PS92]; we call them TF binding motifs (TFBMs). In the case of microbial organisms there exists five platforms [BTR09] which store information about predicted and experimentally validated TFBMs. They are RegulonDB [GCJJPG+08], MtbRegList [JGC+05], PRODORIC [MHB+03], DBTBS [SMdHN08], and CoryneRegNet [BWR+07]. These platforms hold interesting information concerning large-scale transcriptional networks reconstruction. From the information they contain, several tools and analyses are derived. Some of them are included as functionalities in the platforms. We may divide them in three categories:

1. **Network information organization:** including databases storing all the regulatory information. Also tools providing a web interface, navigation capabilities, and web applications to browse in the genome information of interest (*e.g.* a TF and the region in the genome where it binds to activate other genes). Software for network visualization and exchange of information is also included in this category.

2. **Sequence analysis to discover regulations:** we include in this category tools and methodologies for the analysis of sequences in order to better represent the regulatory knowledge. We also include network structure topological analyses. Methods in this category may perform the following tasks:

   - Prediction of new network regulations by using tools that can predict new TFBMs, promoter sequences, and TFs.
   - Prediction of the role of a TF by exploring the transcription start sites.
   - Prediction of gene operons.
   - Prediction of homologous genes or proteins.

- Topological studies of the organization of regulatory networks. These studies develop a series of statistic measures in order to elucidate, for example, hierarchy or connectivity notions of a system [FGAPTQCV08b]. Most of the time they do not require neither gene expression data nor notions about the role of the regulation (activator, repressor).

3. **Network and/or microarray analysis:** we include here methods devoted to the analysis of a regulatory network using biological knowledge, microarray data, or other type of high-throughput methodologies. This category of methods is extensive; we can, however, subdivide it by the type of results generated into:

   (A) Methods that formalize the network interactions using a mathematical model, and thus perform simulations or reasoning analyses (see Section 1.2).

   (B) Methods that integrate high-throughput data into regulatory networks (see Section 1.3).

   (C) Methods that classify extensively high-throughput data and TFBMs information. This classification may lead to generation of the new knowledge related to the topology of the network [EBK+08]. Also, it may lead to better analysis of the fuzzy microarray outputs and provide a better biological interpretation of them [RZD+07].

   Generally speaking, methods in (A) and (B) are mostly centered on the network, while methods in (C) on the data. Methods in (A) are able to perform precise simulations of the dynamics of a real regulatory network, in most of the cases a small-scale one. They are not able, however, to use the wide range of post-genomic data. On the other hand, methods in (B) relax the strict formalization of the model of regulations in order to include wide genome datasets.

Bioinformatic tools and methods are essential in designing integrated biological knowledge from the large amount of data issued after the genomics era. We can classify those related to the study of regulatory networks in the three aforementioned categories. The research we present in this thesis concerns methods in category 3. Methods provided in categories 2 and 3 may generate hypotheses to be validated in future wet lab studies. In addition, the outputs of methods in category 2 and 3 are connected. For example, methods which study specific network motifs (category 2) are related to the study of the dynamics of the whole system (category 3).

Of the five platforms of large-scale microbial transcriptional networks, RegulonDB is the one containing the major electronically encoded regulatory network of a free-living organism - the *E. coli* bacteria. Thus, many bioinformatic resources are linked to this platform [CVSM+09]. However, only one of the five platforms (CoryneRegNet) includes a tool that automatizes the analysis of large-scale networks and microarray data [BA08]. This is the first indicator of the difficulty of this task for large-scale networks. In fact, the inconvenience has mostly two reasons: *(i)* the difficulty of the network reconstruction

process, as methods of category 3 can only be applied after a regulatory model is built, and *(ii)* the difficulty of automatic reasoning over a large-scale regulatory model.

1.1.3 Network visualization approaches

In addition to the platforms managing information from the large transcriptional microbial networks, and to the large number of existing methodologies that formalize the analysis of regulatory data and reason over it, there also exist some informatic solutions destined to present network analysis in a user-friendly manner. The exchange of information between different fields is fundamental for the progression and the coherency of the results obtained in systems biology. Hence, all the efforts made on diffusing, in a comprehensible way, these results are of great interest. We can cite among these efforts the Cytoscape software package [CSC+07], which is a free and open-source software that provides a framework for the visualization, modeling, and analysis of regulatory networks. Many Cytoscape plugins for network analysis were proposed to this date (see Table 1.1). As already shown, the network analysis concept is too wide. Moreover, the analysis plugins proposed, though centered mainly on the network, share very few properties. Of the 29 analysis plugins proposed, 6 provide topological analyses of the network without considering experimental data [RRAS08, ARS+08], and 7 provide clustering and classification of subnetworks from expression data [YWHT08, CLL+07]. Only 3 plugins propose automatizing network analysis based on a mathematical representation of the regulatory network and experimental data [BA08, GBMS09]; of them only 2 provide a solution in the case of TRNs.

In addition to Cytoscape, there also exist other informatic tools that simulate the network dynamics. We can cite among them Cell Designer [FMKT04]. This tool is a diagram editor for gene regulatory and biochemical networks coded in SBML (Systems Biology Markup Language) [HFS+03], a standard way of representing biological networks.

1.2 Modeling regulatory networks - dynamics formalization

The study of regulatory systems control benefited from the explosion of high-throughput technologies that measure simultaneously thousands of expression levels of cell molecules, *e.g.* cDNA microarrays [BB99], oligonucleotidechips [LFGL99], SAGE analysis [VZVK95], and protein microarrays [MS00]. These studies intend to elucidate the dynamics of a system or analyze the network connections by exploiting the information outputted by the experiments or biological literature. We will review in this section the different kinds of formalisms approached.

Notice, however, that we only consider in this review methods that study transcriptional and signaling networks. The interactions in these networks represent cause-effect relationships, *i.e.*, the edges of the network represent effects of one molecule over the other. There exist, however, other types of genetic regulatory structures where gene

Table 1.1: Cytoscape plugins for network analysis, extracted from [cyt] on July 2009.

Type of analysis	**Number**	**Cytoscape Analysis Plugins**
Topological	6	CalculateNodeDegree, CentiScaPe, CyOog [RRAS08], cyto-Hubba, netMatch, NetworkAnalyzer [ARS+08]
Clustering and classification	7	clusterExplorerPlugin, clusterMaker, jActiveModules, MCODE, TransClust, NetAtlas [YWHT08], PinnacleZ [CLL+07]
Mathematical model	3	Coma [BA08], PerturbationAnalyzer, BioQuali [GBMS09]
Network operations	2	ShortestPath, RandomNetworks
Visualization	2	dynamicXpr, ReOrientPlugin
Validations of network interactions	2	APID2NET [HTPDlR07], CABIN [SD07]
Filtering of network attributes	2	EnhancedSearch, HiderSlider
Others	5	RDFScape (ontologies), BLAST2SimilarityGraph (sequences), OmicsViz (orthologous genes), structureViz (3D protein structure), VistaClaraPlugin [Kin04] (multi-experiments visualization)

connections represent the presence of protein-protein interactions, or the cell viability: synthetic-lethal genetic interactions [DTC+05], or the fact that these genes code for enzymes that catalyze chemical reactions: metabolic networks. In this review we are excluding methods that analyze such types of networks.

1.2.1 Bayesian models - inferring regulatory networks

In the Bayesian formalism regulatory networks are modeled by defining a probability distribution over the expression of a gene. For each gene in the network a probability of its expression is defined, which depends on the probability of the expression of its predecessors in the network. Techniques for Bayesian networks [Hec98, FLNP00] allow us to infer the network topology from a gene expression dataset, obtained for example from microarray measurements. The reconstructed network is defined as the most likely model given the data. Such an optimization problem is usually non-convex, and finding a global optimum cannot be guaranteed in practice. Existing algorithms report a local optimum which should be interpreted with care: errors can appear, and no consensual model may be produced. As an illustration, special attention was paid to the reconstruction of the *S. cerevisiae* network from ChIP-chip data and protein-protein interaction networks [LRR+02]. The first regulatory network was obtained with promoter sequence analysis methods [HGL+04, MWG+06], yet, some undetected transcriptional regulatory motifs were proposed using non-parametric causality tests [XvdL05]. Bayesian analysis also identified new regulatory modules for this network [SSR+03, NTIM05]. However, the results obtained with the different methods did not coincide and a fully data-driven search is in general subject to over-fitting and not fully reliable [FMS+07].

One of the advantages of this formalism is that it can be easily adapted to the noise

present in microarray measurements; however, it does not illustrate the dynamics of the system, nor a reasoning logic confronting the regulatory network and the experimental data.

1.2.2 Boolean networks and generalized logical networks

Boolean networks were introduced in the 70. by Kauffman [KPST69]. In this formalism genetic regulatory networks are modeled as qualitative systems of discrete on/off variables. A Boolean network consists of binary (on/off) genes, interactions that are causal or regulatory links between genes, and rules that specify the state at time $t+1$ of an activity of a gene as a function (output) of the activities of other genes at time t. One can study the dynamics of a Boolean network by following the trajectories of the on/off values for the genes at different time points. One of the advantages of this approach is the possibility to study network properties for large-scale regulatory systems [Kau93]. In [HSWK02] the authors modeled real transcriptional networks with Boolean rules extracted from experimental data, comparing their properties with respect to a model using random Boolean rules. The authors deduced that for 150 small-scale transcriptional systems, where mutational and deletion analysis were provided, eukaryotic genes had a strong bias to be regulated with an OR Boolean function. In [KPST03] global properties of the *S. cerevisiae* transcriptional network were studied, deducing also a strong bias for OR Boolean functions, and a more stable behavior than for random networks. This formalism idealizes transitions between activation states as deterministic and synchronous. The synchronicity assumption is quite strong for biological reality because the effect of a gene resulting from the change of activity of one of its regulators could be seen after several time points. When transitions are not simultaneous, certain behaviors may not be predicted.

Boolean network models evolved into generalized asynchronous logical networks [Tho91]. The concentration of a gene in the network is represented no longer as a binary variable, but as a logical variable: a discrete value from 0 to p, where p is the number of successors of the gene in the network. The network interactions represent positive or negative influences labeled with a discrete value that represents the threshold of activity of the interaction, *i.e.* the interaction will take place only if the concentration (logical variable) of the source gene overpasses the value of the interaction label. In addition, a logical function, that takes as input the logical variables of the predecessors of a gene, outputs the state which the gene tends to be in a future step. When the system does not change its state, it has arrived to a steady state. Using this formalism, it is possible to find all the steady states of the system. Under appropriate parametric ranges, positive circuits generate multistationarity, while negative circuits generate homeostasis [TTK95]. By detecting the feedback circuits on the system and analyzing its steady states, global properties of the system can be obtained, as shown in [MTAB99] for the study of the regulatory network of *Arabidopsis thaliana*. More recent applications of this approach are the analysis over the small-scale biological networks of the the embryonic development of the drosophila [GCT08], and the development of the cystic fibrosis [GMBC$^+$04].

1.2.3 Quantitative and qualitative differential equations

Ordinary Differential Equations (ODEs) appear as an alternative to the discrete modeling introduced above. In this formalism, the concentrations of the network products are modeled by continuous time-dependant variables. Network components represent different biomolecular components as genes, proteins, and metabolites. Gene regulation is modeled by the rate equations expressing the rate of production of a network component as a function of the concentrations of other components. Rate equations include constant rates, kinetic parameters, and regulatory functions. They express a balance between the number of molecules appearing and disappearing each time unit. The regulatory function representing the production of a gene is given by the Hill (continuous sigmoid) curve [YY71].

As a result, regulatory networks are modeled by a system of rate equations. The solution of this system is approximated by numerical techniques that allow further analysis and simulation. Bifurcation analysis tools [Str94] may limit the choice of parameters. In [TCN01] the authors applied these tools to exploit a numerical model of the cell cycle. ODEs were also applied to model the signaling pathways of NFκB [NIE$^+$04] and EGF [SEJGM02]. The advantage of this formalism is that it takes into account different types of molecular interactions and gives very precise results on the system dynamics when used on a known small-scale regulatory network.

However, for many biological processes detailed quantitative information is not available. Hence, qualitative formalisms appeared, which study the qualitative dynamics of the system. Qualitative reasoning was invented by Kuipers in 1986 [Kui86] and it was applied in medicine and physics, as well as in molecular biology. There are several tools for simulating qualitative reasoning; one of them is QSim [Kui94], in which a model is described in terms of at least one qualitative differential equation (QDE). Notice that a QDE consists of a set of variables, a set of quantity spaces for each variable, a set of constraints for the variables, and a set of transitions. Each variable has a magnitude (numerical value) and a direction (increasing, decreasing, or zero). In [HSK98b] the authors used QSim to qualitatively simulate the λ phage infection in *E. coli*. The same authors developed BioSim [HSK98a], which is an improved simulator of qualitative reasoning. Genetic Network Analyzer (GNA; [dJGHP03]) is another qualitative simulator, applied in the simulation of the initiation of sporulation in *B. subtilis*. It represents regulations among genes not by using the continuous Hill functions, but by discontinuous step functions, building in this way a system of piecewise-linear differential equations. Recent illustrations that used GNA to simulate the qualitative dynamics of more complex, but still small-scale, regulatory networks are the studies of the nutritional stress in *E. coli* [BRdJ$^+$05], and of the onset of a pectinolytic bacterium [SRN07]. Interestingly, GNA also proposes model checking techniques to analyze the model behavior. In this way, the analysis of regulatory networks considers not only the dynamics of a system, but also it takes into consideration certain model properties.

1.2.4 Rule-based formalisms - signaling networks

The analysis of regulatory networks was approached first from the dynamics simulation perspective; afterwards, with tools as Genetic Network Analyzer, the system behavior was also explored. The system behavior analysis can be formalized with the rule-based approach. We will introduce this formalism, by first describing the type of regulatory network which fits the best to this approach: signaling pathways.

Signaling pathways are a more complex type of regulatory networks. Nodes in these networks represent proteins in different states *e.g.* phosphorylated, unphosphorylated, forming a complex, *etc.* Interactions represent not only transcriptional regulations, but also post-translational modifications of proteins, which in turn generate a large number of possible molecular species that can carry the signals. Compartments and transport phenomena may also be represented in these networks. Analyzing this type of networks requires at first a careful understanding of the biological process, and then a mapping of this knowledge into regulatory constraints.

Kappa [DL04] is a formal language for molecular biology where precise rules of post-translational regulations can be stated. Thus, biological systems can be represented using Kappa, and the properties of the program built from these rules can be analyzed statistically [FDK$^+$09]. By interpreting rules into ODEs, it is possible to simulate K-rules programs in order to obtain the dynamics of products in the network. In [DFF$^+$07] authors showed the performance of this rule-based formalism on a system of 10 components that generated 10^{23} possible molecular states for the network components. Despite the number of states, they were able to generate the dynamics of the system. The advantage of this modeling approach, with respect to the formalism of ODEs, is that it does not assume synchronicity in cell dynamics and that it can calculate the dynamics for systems with a large number of possible states of the species. A difficulty in the ODEs simulation proposed by this approach is that it needs to fix the rate constants of the system. As the rules are very precise, the rate constants need to be known *a priori*. Besides, as we shall see in the following chapters, signaling networks can be built precisely for 100 components. The authors of this formalism do not specify its performance in terms of computation time when referring to middle or large-scale signaling networks.

Another rule-based formalism is BIOCHAM [CFS06, FSCR04]: a language and programming environment for precise modeling biological networks and formalizing of the experimental knowledge as properties of the system. The provided rule-based language allows modeling biological networks at three abstraction (semantic) levels: (1) Boolean, where the molecules of the network are represented by their present or absent value and the interactions are represented by a set of rules, (2) concentration, where the variables denote real values expressing the concentration of the molecules and the rules are equipped with kinetic expressions that provide the dynamics of the system based on ODEs, and (3) population, where the modeling object is an integer number representing the number of molecules in the system and the rules are interpreted by Markov chains.

The biological properties of the system are formalized using temporal logic CTL (Computation Tree Logic), LTL (Linear Time Logic), and PLTL (Probabilistic LTL), used in the three semantics respectively. The outputs of BIOCHAM are: model validation wrt quantitative or qualitative specification, simulations for quantitative data given an initial condition, and model correction and parameter inference wrt global properties that are given in terms of constraints representing known biological behaviors of the system. These properties correspond to the biological experiments used to build and validate the model. The verification of biological properties was implemented in Prolog for the numerical LTL properties [CCRFS06], and through the NuSMV modelchecker for the CTL properties [CCG+02]. The largest example treated with the Boolean semantics was a middle-scale model of 800 rules and 500 variables; its performance was shown in [CRCD+04] to be of a few tenths of seconds to compile the model and check simple CTL formulae. One of the limits of the Boolean semantics is the high degree of nondeterminism: many paths are possible, which leads to performance problems on small size models having a high number of parallel pathways. In the population semantics only small-scale networks can be treated. Temporal logic is an interesting approach to reason automatically over time on very precise signaling networks, allowing the user to perform analyses over qualitative and quantitative data. It is fundamentally conceived to reason over the temporal combinatorics of the system, being its main interest the evolution of the system through time.

1.3 Confronting gene expression levels with large-scale regulatory networks

As reviewed in Section 1.2, the quantitative study of the dynamics of a regulatory system in terms of attractors, steady states, and simulations is limited by the system size. Larger systems require larger knowledge of them; and even in the case of QDEs, the necessity of reviewing literature to find all the applicable thresholds of interactions may pose a big problem when considering a network of a thousand of products. Besides, as mentioned in Section 1.1.2, large-scale TRNs have already been compiled for bacteria. In the case of eukaryotes, a reasonably complete picture of the architecture of *Saccharomyces cerevisiae* TRN was proposed by [NGBBK02, TMJ+06, MMT+08]. This rises two natural questions. The first one, "How feasible can be to integrate the large-scale compiled regulatory knowledge into a methodology or bioinformatic tool, to simulate the network or reason over it?" The second one, "How correct a comparison between the outputs of such methodology with genomic datasets can be, knowing that networks, even after being curated, are always incomplete [EBK+08], and that genome-wide datasets may be uncertain [GW02]?"

Recently, large-scale network analysis methods appeared to handle the complexity of large-scale networks. These methods leave aside the study of the network dynamics to analyze less precise and more abstract properties in large-scale networks. One of the widely studied properties is the *consistency* of a large-scale network wrt gene expression datasets. This property, which will be discussed through all this thesis, re-

ports how coherent are the experimental observations wrt the network topology, that is, wrt the causal arrows in the network. Once the consistency diagnosis performed, one checks whether an experimental dataset fits the regulatory model or not; also, one checks which regions of the network are incomplete wrt a specific dataset. This concept was approached by different methods which will be discussed in the following paragraphs. Some of them propose to decompose the large network structure into network motifs or building blocks. Others generate qualitative computational predictions of gene-expression. The common point between these approaches is that their reasoning is based on the use of causal rules to represent the regulatory interactions of the network.

The work of [GRRL$^+$03] used the Boolean networks formalism as well as a careful review of the literature to provide a partial consistency measure for the *E. coli* regulatory network using microarray profiles. Its novelty resided in the ability of widely exploit genome expression profiles to evaluate their consistency with known regulatory structures. The authors tested the consistency between the *E. coli* regulatory model proposed by RegulonDB and four microarray profiles on this organism designed by them. The network structure was analyzed partially, considering only specific regulons (genes that are controlled by the same TF) in which genes are expressed homogeneously in the microarray. A regulon was marked as consistent when the on/off expressed values of its genes agreed with the role of the TF that transcribes them and its activity. The activity of the TF was measured taking into account its DNA binding conformation, which was deduced also from the expression of the genes only regulated by it. A disjunctive Boolean rule was used to model co-regulated groups except in the case where literature suggested other type of Boolean regulation. The results revealed that 70 to 87% of the network regulons analyzed were consistent with the four experimental settings. As expected, they obtained better consistency results when using the DNA binding conformation than when not.

Following the same direction, the authors in [HLPP06] also evaluated the consistency between two network structures and a compendium of gene expression profiles. They used the networks of *E. coli* [SOMMA02] and *S. cerevisiae* [NGBBK02]. The datasets were composed of 163 experiments for *E. coli* and 904 for yeast. They divided the network regulations into five types of motifs: regulons, complex regulons (genes receiving more than one activator or repressor), regulator-target interactions (RIs), target modules (TMs, multiple regulators acting over the same target), and feed-forward loops. Afterwards, they evaluated the consistency of a motif in terms of how probable was that its topology reflected a coherent gene expression tendency, when compared to random generated motifs. For example, the genes in a regulon were supposed to correlate in their expression profiles significantly (P-value < 0.01) wrt the expression profiles of random generated regulons. As a result they obtained different percentage of consistent motifs according to the motif type. In the case of regulons they obtained that around 50% of them were significantly consistent in both yeast and *E. coli*. The lowest percentage of consistency was obtained for RIs. These analyses revealed incomplete regions in the model. Interestingly, a different number of consistent regulons was

obtained for regulons composed of activators than of repressors, and for TM motifs with different number of regulators. The difference between this approach and the previous one is that no formalism is applied over the model, and the consistency of regulatory rules are deduced from a significant pattern of expression among the different expression profiles. This analysis gives a strong confidence to datasets, and makes difficult to detect experiments in which data points are ambiguous or incoherent. Also, a weakness relies on the fact that much information is abstracted in order to generate a single measure for consistency.

Another approach aimed to analyze large-scale networks is the work of [CKR+04]. The authors applied a procedure described in [CSP01] to integrate high-throughput and computational data to elucidate large-scale bacterial networks. The procedure was able to predict qualitative gene expression events by using detailed metabolic information of the model. It expanded flux-balance analysis [VP94] to include Boolean regulatory constraints. The presence or absence (1/0 value) of a protein in the network was deduced from the quantitative predictions of the metabolite concentrations when the bacteria growth was optimized. Authors could simulate the growth of the organism by calculating, at first, the optimal metabolic flux distribution using fixed numerical parameters and a set of values as initial condition. Then, they iteratively calculated the next step of the system by using the resulting flux distribution and the conditions of the system at the previous step. In this way, on/off values of proteins were obtained for each time point, and the up-, down-regulation, or zero tendency were deduced for all the proteins involved in the model. In [CKR+04] this method was applied to the large-scale *E. coli* metabolic/regulatory network, obtaining initially a 49% of accuracy between their gene-expression computational predictions and the mRNA measurements during the aerobic-anaerobic shift. This percentage of accuracy was, not surprisingly, increased to 98% when the regulatory boolean rules where modified according to the experimental outputs. The same authors recently integrated signal transduction models into their *in silico E. coli* computational model in [CXCK08]. Their integrative approach is relevant because it combines and analyzes biological regulations occurring at different levels in an organism. This can, however, raise problems when the interactions among the organism molecules are not precisely described. One weakness is that obtaining gene expression predictions requires previous knowledge on the metabolite initial concentrations and reactions of the organism. Additionally, to simulate different time steps they need to know (or impose) numerical parameters on the protein transcription and decay time, as well as maximum uptake rates for all the possible substrates. Even though simulations are performed automatically, the authors of this approach do not mention computation time of their procedure. In comparison with the previous studies, it can also be seen as a method that fits the regulatory rules into an specific dataset.

In this section we discussed methods that reason over large-scale regulatory networks. For that purpose some of them decompose the network structure into network motifs [GRRL+03, HLPP06]. Others use the Boolean networks formalism [GRRL+03, CKR+04] to compute a consistency measure of the network regulations according to the

experimental outputs, as well as to generate qualitative gene expression computational predictions. The main interest of these approaches is that they proposed a direction for analyzing large-scale regulatory networks. However, the proposed consistency analysis was neither automatized nor global. Generating computational predictions can be automatized, yet it depends on acquiring precise quantitative metabolic knowledge of the system.

1.4 Large-scale qualitative modeling

The studies discussed in Section 1.3 dealt with genome-scale datasets and large-scale networks. Those that modeled the networks using the Boolean network formalism reasoned with logical rules on variations of products during stress experiments or gene perturbations. The main idea underlying these approaches was to set up constraints on variations between different states. This point of view provides a new application of the original Boolean rules of [Kau93] in a considerably modified context that ensures their validity: variations during shifts instead of dynamical simulation.

Based on this idea, in [SRB+06] a formal method was proposed to investigate the global consistency of a regulatory model with large-scale differential gene expression data. A general interaction logical causal rule (near to a logical disjunction) was used to addressed the question of consistency.

The work in this thesis concerns the biological application and computational extension of this approach: initially, to test its validity on large-scale bacterial transcriptional networks, afterwards, to apply it to complex eukaryotic signal transduction models. During this work, the original approach was adapted to deal with more complex regulatory interactions, especially when dealing with eukaryotic data. Also, it was extended to deal with knockout and over-expression experimental datasets. Three types of output were constantly checked, adapted, and improved in all our case studies:

- Global consistency of the regulatory network with independent gene expression datasets.
- Diagnosis of the inconsistent modules.
- Predictions over gene expression or protein activities.

These outputs were generated after running complex algorithms, which can handle large systems of constraints and find a global solution of the system. A Python library, Bioquali, was used to implement the programs that perform the consistency analysis. Also, bioinformatic tools such as a Website, a Cytoscape plugin, and a Web service application were developed during this thesis and made publicly and freely available. These tools were proposed to prospective biologists so that they can perform the same type of analyses over their own data.

Our work complements previous large-scale reasoning studies since it is automatic and global. We go one step further from the consistency analyses proposed before, as we

provide an automatic tool for repairing inconsistent networks. By doing this we are able to highlight the minimal corrections needed to be made on the network and/or data in order to reconcile them. Contrary to most of the methods described in section 1.3, we do not prioritize the dataset more than the model; we consider both of them prone to errors. In comparison with the results obtained in [CKR+04], our computational predictions, without considering metabolic knowledge on the system, were highly accurate (90%) after automatic retrieving of the inconsistencies of a large-scale transcriptional network with independent datasets.

On the whole, our results answered to relevant biological questions such as:

- Testing the global consistency of large-scale and complex biological networks wrt to high-throughput datasets.
- Generating manual and automatic hypotheses to solve the inconsistencies between a network topology and experimental data.
- Inferring the role of transcription factors as activators or repressors.
- Globally and extensively reasoning over the on/off states of the paths connecting two interesting nodes in a signaling network.

Being the domain of this thesis multi-disciplinary, the proposed solutions and methods answer at different levels methodological, bioinformatic, and biological aspects. This thesis is divided in two parts. Part I describes the mathematical and informatic methodology used to approach the presented problem, as well as the bioinformatic tools proposed to diffuse this methodology. Part II presents the biological application and validation of our results.

Part I

The approach used in this thesis to analyze large-scale regulatory networks is based on a *consistency* measure of a network when confronted to an experimental dataset. Both the network and dataset are represented in a qualitative way, that is the regulations in the network are discretized, as well as the experimental measurements. Afterwards, this discretized information is mapped into a mathematical model composed of qualitative equations. The satisfiability answer of the system of qualitative equations will give us insights about the consistency between the network topology and the experimental data provided. Large-scale regulatory networks can be mapped into large systems of qualitative equations, which solution can only be studied via informatic approaches. Up to this date, two informatic approaches were proposed to deal with this problem, both being very efficient in computation time.

Different programs can be written basing on the implementation of these approaches in order to answer questions related to the confrontation of biological data. We proposed specific programs addressed to answer the following questions: *(i)* consistency check between a network and a dataset, *(ii)* diagnosis of inconsistent network regions

or dataset observations, *(iii)* TF role inference on unsigned networks, *(iv)* automatic computation of all possible inconsistent regions in the network, *(v)* automatic minimal repair of a network and/or a dataset, and *(vi)* post-consistency analysis of predictions in order to find the origin of a molecule change under an experimental condition.

Some of these questions, specifically the consistency analysis and diagnosis, were included into bioinformatic tools that enable the user to access and exploit them in an easier way by using a Website, a network visualization software, or a Web service.

Chapter 2

We present in this chapter the mathematical modeling approach used in this thesis. This approach was initially proposed in [SRB+06] and [RLS+06]. The mathematical model is built from a regulatory network and an expression dataset; it contains qualitative information of both data. The network regulations are originally discretized in $\{+, -, \mathbf{?}\}$ influences, expressing activations, repressions, or complex interactions, while the dataset observations are discretized in $\{+, -\}$ changes representing up- or down-regulations of network products between two experimental conditions. The mathematical model of a network and a dataset is expressed as a system of qualitative constraints, in which the variables of the system are the change in variation of a non-observed network product. Each constraint in the qualitative system relates a node in the network with its direct predecessors using a generic consistency rule, which states the following idea: *"The variation of the concentration level of one molecule in the network must be explained by an influence received from at least one of its predecessors, different from itself, in the network."* If this rule appears to be valid for all the network products, the system of constraints is said to be consistent. In this case, there exists at least one solution for the variations of all network products. The non-observed network products will be fixed to $\{+, -\}$ values in order to satisfy all the constraints of the system. By searching the intersection of all the system solutions we may find the *necessary* changes that have to occur in order to explain the experimental dataset. This intersection is called *predictions* of a consistency analysis. Also, it may happen that the qualitative system of constraints is not consistent with a given dataset. In this case it is possible to search for the cause of this inconsistency; generally, it is a combination of observations in some network products and signs of some network edges. We call it *inconsistent graph*.

We described in a few words the standard analysis of the *consistency check* between a network and a single dataset, which will be detailed further in this chapter. There we will also discuss about new qualitative constraints that were added to the mathematical model in order to answer other types of biological questions, as well as to relate a network product with its direct predecessors in a different way.

Chapter 3

In this chapter we show which informatic solutions lied on many of the results of the research presented in this thesis. Computing the satisfiability of a large system of qualitative constraints is an NP-complete problem for even linear qualitative systems

[Dor88], and classic methods of resolution do not allow solving this kind of problems [TMD03]. Two informatic approaches propose a solution to this problem, and, consequently, they provide an informatic solution to the confrontation between large-scale regulatory networks and experimental data.

The first one is based on the ternary decision diagrams, TDDs [VBSR05, Veb07, LBP07, Bor09]. This approach evolved through time, and it was adapted to provide solutions to different biological questions efficient in computation time. We proposed, basing on the TDDs approach, programs to perform the classic *consistency-check* analysis. This analysis consists of the following steps: (1) finding a solution of the system of constraints, (2) in case a solution is found (consistency), predicting a set of variations of network products, (3) in case there is not a solution (inconsistency), isolating a set of qualitative constraints (inconsistent graph).

From the consistency-check analysis another program was proposed to approximate all inconsistent graphs. The TDDs approach was also used to design programs which infer the unknown role of TFs by using multiple datasets. Furthermore, two extensions of its implementation were proposed to deal with: *(i)* more complex types of regulatory roles in the network, and *(ii)* expression of the null-variation of a network molecule. While the first extension was applied successfully to signaling networks, the second one is still in its testing phase.

The second informatic approach is reasoning over a qualitative system of constraints using Answer Set Programming (ASP, [GKNS07]). In [GST+08] a program encoding ASP rules was proposed to detect exactly all the inconsistent subgraphs of a network when confronted with a single dataset. Currently, of the proposed programs based on TDDs, only the consistency-check and diagnosis of all inconsistent graphs are implemented and tested in ASP.

Nonetheless, different types of problems can be approached with ASP, for instance, the minimal correction of a network/dataset when they are inconsistent. The program containing the ASP rules to answer this question is presented in detail in this chapter. This work was validated using the *E. coli* TRN and was submitted, in collaboration with the team leaded by T. Schaub in the Potsdam University, to the "Twelfth International Conference on the Principles of Knowledge Representation and Reasoning", to be held in Toronto, Canada, May 9-13, 2010.

Chapter 4

The informatic implementations used to analyze large-scale networks offer a good starting point for users with specific biological questions and basic computer science knowledge. Both implementations propose a framework (either by coding in Python or ASP) in which the user can program her own applications. It may be, however, that users are confronted with the same type of questions we were confronted with, and which we already proposed specific programs to approach the solution for. Thus, in order to provide a faster and easier access to this specific reasoning, we designed three bioinfor-

matic tools. In our opinion, these tools are a way of diffusing these methodologies, by controlling the questions that can be asked to the system and by executing remotely the analyses in high-performance machines.

In this chapter we present these tools, developed during this thesis in order to analyze the consistency of the large-scale regulatory networks and diagnose them wrt a single dataset. All of them launch their analysis on the GenOuest high performance computing facility [gen], are publicly available, well documented, and constantly maintained. They were set up in collaboration with the GenOuest platform.

The first one consists of a Website which receives text files representing a network and a dataset; it outputs the consistency answer, the predictions, and lists the network regulations causing an inconsistency.

The second tool is a plugin for Cytoscape [SMO+03], a software for visualizing and analyzing regulatory networks. This plugin allows visualizing the consistency-check analysis, as well as the diagnosis of the inconsistencies. An automatic detection of different network inconsistencies is also proposed, in addition to user-friendly conversions of network annotations and experimental results into discretized values. The work concerning the Cytoscape plugin was published in "Guziolowski *et al.* BioQuali Cytoscape plugin: analysing the global consistency of regulatory networks. BMC Genomics, vol. 10, pp. 244, 2009 ".

The third tool proposes the same consistency analysis, but this time as a Web service. Thus, users can automate access to the consistency-check analysis from their own informatic tools or databases. The Web service was developed in order to interact with the CoryneRegNet platform [BWR+07], which is a systems biology platform to analyze gene regulatory networks in *Corynebacteria* and *E. coli*. It could complement existing tools of this platform, since it is able to perform global and thus more complex analyses in reasonable computation time.

Part II

The mathematical and informatic approaches introduced in the previous chapters were validated using real biological data. On this account we obtained from public databases and from the literature three large-scale regulatory networks of different organisms. In addition to this data, we also collected experimental data of gene expression profiles on these organisms.

The first model corresponded to the *E. coli* transcriptional regulatory network obtained from the RegulonDB database. The second one was the *S. cerevisiae* transcriptional network obtained from ChIP-chip data. The third one was the signaling network of the EWS-FLI1 human oncogene. This last network was obtained from a collaboration with the Institute Curie[1] [SZN+08].

We compiled our results into two chapters, 5 and 6, concerning analyses on prokaryotic and eukaryotic data, respectively.

[1] http://bioinfo-out.curie.fr/projects/sitcon/

Chapter 5

The bacterium *Escherichia coli* is one of the best-studied single-celled organisms. Its genetic regulatory network was studied by a wide community, and this information was compiled in databases such as RegulonDB. For that reason, we applied the modeling and informatic approaches to this bacteria. We illustrate in this chapter the biological application of the qualitative consistency analysis on this organism.

First, we validated the computational feasibility of the consistency-check analysis on the signed *E. coli* TRN. Our analysis was performed in less than one minute using programs based on the TDDs approach. As a result, we completed the network of regulations and proposed corrections on the dataset of observations. We published this work in "Guziolowski *et al.* Checking Consistency Between Expression Data and Large Scale Regulatory Networks: A Case Study, Journal of Biological Physics and Chemistry, vol. 7, pp. 37-43, 2007".

Afterwards, we validated the accuracy of the computational predictions by comparing them to an independent microarray dataset. We detected that 80% of the computational predictions were in agreement with the independent dataset observations. We also proposed explanations of our false-positive predictions. This work was published in "Guziolowski *et al.* Curating a large-scale regulatory network by evaluating its consistency with expression datasets, CIBB'08: 5th International Conference on Bioinformatics and Biostatistics, Salerno, Italy 2008, Lecture Notes in Computer Science, vol. 5488, pp. 144-155, Springer".

The *E. coli* TRN was also used to asses the automatic correction of inconsistencies when using genome-wide datasets. We showed that after an automatic correction of the inconsistencies, we can predict partial observations in these datasets with a 90% of accuracy. Finally, we applied the TF role inference approach to the *E. coli* unsigned TRN and validated our results. Using this network we also computed the theoretical limits of the TF role inference approach.

Chapter 6

In the previous chapter we applied our modeling approach to the *E. coli* transcriptional network. Although this organism is very well studied, its transcriptional regulatory machine is simplified with respect to eukaryotic organisms. One major difference between prokaryotes and eukaryotes is the existence of the nucleus membrane, which divides the place where transcription and translation occurs in eukaryotic cells. Another difference is that in eukaryotic cells much more post-transcriptional processes take place after the mRNA production. In prokaryotic cells mRNA usually stays unchanged.

In this chapter we will discuss the results obtained when dealing with two eukaryotic regulatory network models. The first one is the transcriptional regulatory network of *S. cerevisiae*. The second one is the signaling network of the EWS-FLI1 human oncogene. We present the different analyses proposed over these more complex systems, as well as the biological impact and validation of our results.

In our first study we show a comparison between the TF role inference approach when applied to the *S. cerevisiae* eukaryotic regulatory model with respect to the *E. coli*

prokaryotic model. We published this work in "Veber *et al.* Inferring the role of transcription factors in regulatory networks, BMC Bioinformatics, vol. 9, pp. 228, 2008".

In our second study, we describe how we adapted our approach in order to consider specific representations of post-translational phenomena in the EWS-FLI1 signaling network. Based on this more specific modeling, we studied the pathways responsible for the inhibition and reactivation of the cell-cycle phenotype in this network. We submitted this work to the IEEE/ACM "Transactions on Computational Biology and Bioinformatics" (IEEE-TCCB) journal.

Chapter 2

Qualitative modeling approach

In this chapter we describe the mathematical formalism of regulatory networks, which all the results of this thesis are based on. We present three different types of problems that can be approached with it. For each problem we detail the biological data that is accepted, as well as which type of outputs can be generated. All problems are stated by a set of qualitative constraints, which are specific to the type of data we are treating; we will see how we adapted our modeling to handle different levels of biological knowledge.

2.1 Mathematical formalism

2.1.1 Intuitive consistency rule

Let us first introduce intuitively the consistency criteria used in this approach. A regulatory network is said to be *consistent* if the following sentence, called *generic consistency rule*, holds for all the molecules in the network: *"The variation of the concentration level of one molecule in the network must be explained by an influence received from at least one of its predecessors, different from itself, in the network".*

The biological intuition of the consistency rule can be viewed in Fig. 2.1. A and B may represent two proteins that activate the transcription of gene C. The consistency rule states that if A and B are both up-regulated (+) under certain condition, then C must be up-regulated, *i.e.* a '+' *prediction* will be assigned to C (Fig. 2.1a). Similarly, the concentration change of C will be predicted as '−' if both, A and B, were down-regulated.

When A is up-regulated (+) and B is down-regulated (−) the expression level of C cannot be *predicted*, as both expression levels (up/down regulated) are possible for C and do not contradict the consistency rule (Fig. 2.1b).

A third situation may occur when all the molecules are observed, let us say, A is up-regulated, B is up-regulated, and C is down-regulated. The consistency rule states that C should be up-regulated; the experiment, however, shows the contrary. Thus, we arrive to an *inconsistency* between the network and the experiment, also called *inconsistent graph* (Fig. 2.1c). No predictions may be generated from an inconsistent graph, yet, a

region in the network is identified together with the expression data that created the conflict.

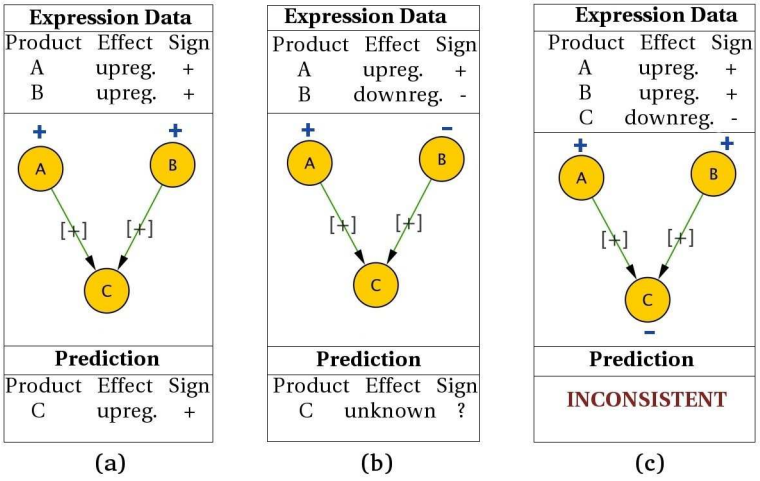


Figure 2.1: Examples explaining the consistency check process. **(a)** Expression predictions when a consistent expression dataset is provided. **(b)** A consistent expression dataset may not generate a new prediction. **(c)** An expression dataset provided was inconsistent with the influences in the graph.

A regulatory network is ***consistent wrt an expression dataset*** if all the qualitative expression changes, observed in the dataset, are explained by the (consensual or not) fluctuation of some of the nodes in the network. In Fig. 2.1 we see that depending on the expression dataset we may obtain up to three different results: consistency and prediction, only consistency, or inconsistency.

2.1.2 Formalization

The mathematical formalization of the consistency rule is applied on a regulatory network represented in the form of an ***influence graph***. An influence graph is a common representation for biochemical systems where arrows show activations or inhibitions. Basically, an arrow between A and B means that an increase of A tends to increase or decrease the production rate of B depending on the shape of the arrow head. For example, the influence graph of the network describing the lactose metabolism in the bacterium *E.coli* (lactose operon) is shown in Fig. 2.3. Common sense and simple biological intuition can be used to state that an increase of allolactose (node A on Figure 2.3) should result in a decrease of the production rate of $LacI$ protein. However, if both $LacI$ and $cAMP$-CRP increase, then nothing can be said about the variation of $LacY$.

Let us now generalize the influence graph concept in the mathematical study of the ***equilibrium shift*** of a differential system. On that account, let us consider a network of n interacting cellular constituents (mRNAs, proteins, or metabolites). We denote by X_i the concentration of the i^{th} species, and by $\mathbf{X}$ the vector of concentrations (which

components are X_i). We assume that the system can be adequately described by a system of differential equations of the form $\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}, \mathbf{P})$, where $\mathbf{P}$ denotes a set of control parameters (inputs to the system). A *steady state* of the system is a solution of the system of equations $\mathbf{F}(\mathbf{X}, \mathbf{P}) = \mathbf{0}$ for fixed $\mathbf{P}$.

A typical experiment consists of applying a perturbation (change $\mathbf{P}$) to the system in a given initial steady state condition $eq1$, waiting long enough for a new steady state $eq2$, and recording the changes of X_i. Thus, we shall interpret the sign of DNA chips differential data as the sign of the variations $X_i^{eq2} - X_i^{eq1}$.

The particular form of vector function $\mathbf{F}$ is unknown in general, but this will not be needed as we are interested only in the signs of the variations. Indeed, the only information we need about $\mathbf{F}$ is the sign of its partial derivatives $\frac{\partial F_i}{\partial X_j}$. We call *influence graph* the graph which nodes are the constituents $\{1, \ldots, n\}$, and where there is an edge $j \to i$ iff $\frac{\partial F_i}{\partial X_j} \neq 0$ (an arrow $j \to i$ means that the rate of production of i depends on X_j). As soon as $\mathbf{F}$ is non linear, $\frac{\partial F_i}{\partial X_j}$ may depend on the actual state $\mathbf{X}$. In the following, we will assume that the *sign* of $\frac{\partial F_i}{\partial X_j}$ is constant, that is, that the influence graph is independent of the state. This rather strong hypothesis can be replaced by a milder one specified in [RLS$^+$06, SRB$^+$06] meaning essentially that the sign of the interactions do not change on a path of intermediate states connecting the initial and the final steady states.

To study the equilibrium shift of a regulatory network we need two main elements: *(i)* an influence graph representing the regulatory network and *(ii)* a dataset representing shifts of the network products between two conditions at steady state.

Influence graphs can be built using a natural passage from transcriptional regulatory networks. This is because TRNs hold interactions of the form TF-gene, in which the rate of production of the gene is affected by the concentration change of the TF that transcribes it. Even so, any kind of regulatory networks may be studied as long as their interactions can be mapped as *influence* relations, *i.e. A influences B if increasing or decreasing concentration of A affects the production rate of concentration of B*. The influence graph and dataset must fulfill the following conditions:

1. The edges of an influence graph must be labeled qualitatively as $+$, $-$, and **?**, where $+$ represents a positive influence (*e.g.* activation of gene transcription, recognition of a gene promoter region, or formation of proteins complexes), $-$ a negative influence (*e.g.* inhibition of gene transcription, inactivation of a protein), and **?** a dual or complex regulation (see Fig. 2.2). It is important that during the shift from condition $eq1$ to condition $eq2$, the $\{+, -\}$ labels of the network edges remain constant.

2. The dataset must be composed of qualitative $\{+, -\}$ *concentration changes* of some of the molecules in the network. For example, one type of concentration changes may be statistically significant mRNA-expression responses, described qualitatively as: $+$ up-regulation, $-$ down-regulation. We may also provide other reliable concentration changes issued from the literature or other types of stress perturbation experiments.

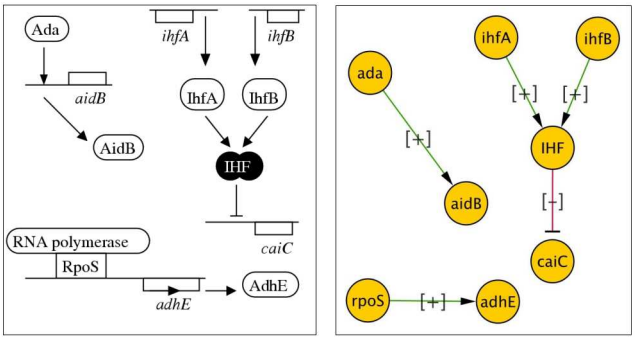


Figure 2.2: A regulatory network (**A**) mapped into an influence graph (**B**). Influences among molecules create an influence graph. The arrows in the influence graph represent a positive (+) or negative (−) influence.

2.1.3 Qualitative constraints

We introduce now an equation that relates the sign of variation of a species to that of its direct predecessors in the influence graph. To state our result with full rigor, we need to introduce the consistency relation '$\simeq$' among signs (see Table 2.1). This relation is reflexive, symmetric, but not transitive, because **?** is consistent with anything.

Table 2.1: Consistency relation $\simeq$. It states the consistency answer of a qualitative constraint. T stands for true, whereas F for false.

$\simeq$	+	−	**?**
+	T	F	T
−	F	T	T
?	T	T	T

Let us now denote as T a node in the influence graph, and as $\{S_1, S_2, \ldots, S_p\}$ the p predecessors of T in the graph, different than T. In addition, let us denote $t \in \{+, -\}$ (resp. $s_1, s_2, \ldots, s_p$) the variation of node T (resp. $S_1, S_2, \ldots, S_3$), measured by comparing two conditions of the cell at a steady state[1]. Recall that we consider experiments that can be modeled as an equilibrium shift of a differential system under a change of its control parameters. In this setting DNA chips outputs, for example, may provide the signs of variation in concentration of many (but not necessarily all) species in the network.

We can thus relate the variation of a node in the influence graph with the variation of its predecessors with the following qualitative constraint:

$$t \simeq F_T(s_1, s_2, \cdots, s_p) \tag{2.1}$$

[1]This notation will be kept through all the formulations in this chapter.

where $F_T : \{+,-\}^p \rightarrow \{+,-,\mathbf{?}\}$ is a qualitative function, representing the combined influence of the predecessors of T. F_T represents how the variations of predecessors of T affect t. In some cases this function may output a **?** value, which means that t cannot be precised. The consistency relation '$\simeq$' is satisfied if a positive variation of a node does not receive a combined negative influence from its predecessors, or vice versa, or if F_T is indeterminate '**?**' (recall Table 2.1).

As we will see in the next sections, F_T can be defined in different ways. It may be precised from the biological literature of the organism under study. Also, it can be generalized mathematically in order to fit any type of behavior of the system under an equilibrium shift.

2.1.4 Generic qualitative constraint

The generic qualitative function $F_T = GEN_T$ is a way to describe the influences of the p predecessors of T in a generic way. It takes into account their regulation sign, that is:

$$t \simeq GEN_T(s_1, \cdots, s_p) \tag{2.2}$$

$$GEN_T(s_1, \cdots, s_p) = \sum_{j \in \{1,\cdots,p\}}^{\oplus} sign(S_j \rightarrow T) \otimes s_j, \ \forall S_j \neq T \tag{2.3}$$

where $sign(S_j \rightarrow T)$, represents the sign of the edge from S_j to T in the influence graph and it is a value in $\{+,-,\mathbf{?}\}$. $\otimes$ and $\oplus$ are the sign operators introduced in Table 2.2. These operators are used to express the total contribution of the predecessors of a node in a graph. GEN_T represents the influences that arrive over a product T in a generic way. If all the contributions $sign(S_j \rightarrow T) \otimes s_j$ targeting T will be positive, GEN_T will be $+$. It is enough that a single contribution will have an opposite sign to obtain $GEN_T = \mathbf{?}$.

Table 2.2: Sign tables for the addition ($\oplus$) and multiplication ($\otimes$) used to build the generic qualitative constraint.

$\oplus$	$+$	$-$	**?**
$+$	$+$	**?**	**?**
$-$	**?**	$-$	**?**
?	**?**	**?**	**?**

$\otimes$	$+$	$-$	**?**
$+$	$+$	$-$	**?**
$-$	$-$	$+$	**?**
?	**?**	**?**	**?**

In [RLS+06] the authors proved that the generic qualitative constraint $t \simeq GEN_T$ is valid under the following hypothesis:

- The molecule T does not self-regulates positively. This condition is verified when T does not activate its own production. Apart from this case, the degradation of a molecule assures that it will self-regulate negatively. If there is a molecule in the influence graph that receives a positive self-regulation, then the generic qualitative constraint is not imposed over it.

- There is not a direct influence from $\mathbf{P}$ (set of system inputs) over T. This means that T is considered as an inner molecule of the system.
- For all $j \in \{1, \cdots, p\}$, $sign(S_j \rightarrow T)$ is constant in the states along the path connecting $eq1$ and $eq2$. This hypothesis implies that the influences of the graph remain constant during the experiment. This is the strongest hypothesis, since it was shown in [ST01] that interactions among molecules may change depending on the concentration level of the molecules influencing their targets.

Notice that the generic consistency constraint is similar to a linearization of the system $\mathbf{F}(\mathbf{X}, \mathbf{P}) = \mathbf{0}$. However, as we only consider signs and not quantities, this constraint is valid even for large perturbations (see [RLS+06] for a complete proof).

In the Boolean networks formalism regulatory interactions in eukaryotes are modeled using an OR Boolean operator, while in BIOCHAM [CFS06, FSCR04] phosphorylation and complex formation phenomena are represented in a constraint using an AND Boolean operator. Remember, however, that in both cases the modeling object is a Boolean variable in $\{0, 1\}$ representing the absence or presence of a molecule in the network at a given time (compared to the previous time step). The modeling object in our formalism represents the qualitative variation between two experimental conditions of a molecule in the network. We do not represent a discrete state in a time point of the molecule, but its qualitative variation between two stable states of the cell. For example, we represent gene up/down-regulations as $\{+, -\}$ qualitative changes of the gene concentration. Nevertheless, Boolean approaches consider up/down-regulations as discrete $\{0, 1\}$ states of a molecule. By using different modeling approaches we model different characteristics of the same biological phenomena.

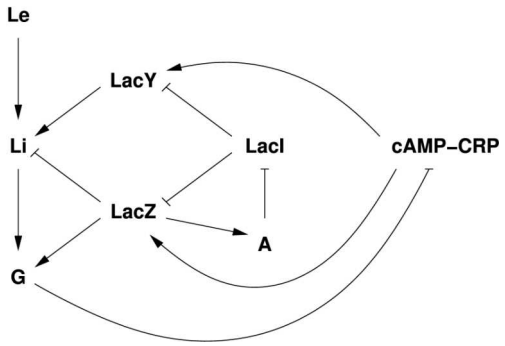
In our case, the generic consistency constraint includes part of the OR and AND operators behaviors. However, when two or more influences carrying opposite signs target the same node, there is not a deterministic behavior of the target node (see Fig. 2.1B). Its change will depend on the priorities of its regulators in governing its expression, which are *a priori* unknown.

2.2 Analyzing a network

2.2.1 Simple example

Let us describe how to reason over a system of qualitative constraints in order to analyze a simple influence graph. Given an influence graph, for instance the graph illustrated in Figure 2.3, we use Equation 2.2 for each node of the graph to build a qualitative system of constraints. The variables of this model are the signs of variation for each species. The qualitative system associated to the lactose operon model is proposed in the right side of Figure 2.3. In order to take into account experimental observations, measured variables should be replaced with their sign values. A *solution* of the qualitative system is defined as a valuation of its variables which does not contain any **?** (otherwise, the

constraints would have a trivial solution with all variables set to **?**) and that, according to the consistency relation $\simeq$, will satisfy all qualitative constraints in the system. If the model is correct and if data is accurate, then the qualitative system must have at least one solution.



$$\begin{cases} LacI & \simeq & -A \\ A & \simeq & LacZ \\ LacZ & \simeq & cAMP - LacI \\ Li & \simeq & Le + LacY - LacZ \\ G & \simeq & Li + LacZ \\ cAMP & \simeq & -G \\ LacY & \simeq & cAMP - LacI \end{cases}$$

Figure 2.3: Influence graph for the lactose operon and its associated qualitative system. In the graph, arrows ending with "−>" or "−|" imply that the initial product activates or represses the production of the product of arrival, respectively. The names of the products correspond indeed to their sign variation between two steady states.

An analysis that can be done is to check the *self-consistency* of the graph, that is to find if the qualitative system without observations has at least one solution. For the lactose operon model the set of all possible variations for its variables takes 2^8 values in $\{+, -\}$. Of them, only 18 satisfy the qualitative system of constraints (see Table 2.3); this set corresponds to the solutions of the system of constraints.

Table 2.3: 18 consistent variations wrt the influence graph model of the lactose operon.

Le	LacI	A	LacZ	Li	G	cAMP	LacY
−	−	+	+	+	+	−	+
−	−	+	+	−	+	−	+
−	−	+	+	−	+	−	−
−	−	+	+	−	−	+	+
−	+	−	−	+	−	+	−
−	+	−	−	+	−	+	+
−	+	−	−	−	−	+	−
−	+	−	−	−	−	+	+
−	+	−	−	+	+	−	−

Le	LacI	A	LacZ	Li	G	cAMP	LacY
+	−	+	+	−	−	+	+
+	−	+	+	+	+	−	+
+	−	+	+	−	+	−	−
+	−	+	+	+	+	−	−
+	−	+	+	−	+	−	+
+	+	−	−	+	−	+	−
+	+	−	−	+	−	+	+
+	+	−	−	−	−	+	−
+	+	−	−	+	+	−	−

Checking the consistency of an influence graph wrt a dataset of experimental measurements boils down to instantiate the measured variables with their experimental value, and to see if the resulting system still has a solution. If this is the case, then it is possible to determine if the model predicts some variations. For example, we can say that the lactose operon model is consistent with the set of observations $\{Le : +, LacI : -, A : +\}$, but not with the set $\{Le : +, LacI : +, A : +\}$.

It happens that a given variable has the same value in all solutions of the system. We call such variable a ***hard component***. The values of the hard components are the predictions of the model.

Whenever the system has no solution, a simple strategy to ***diagnose the problem*** is to isolate a minimal set of inconsistent equations. Note that in our setting isolating a subset of the equations is equivalent to isolating a subgraph of the influence graph. The combination of the diagnosis algorithm and a visualization tool is particularly useful for model refinement as we shall see in Chapter 4.

The resolution of qualitative systems of constraints is a difficult problem, in informatic language it is called an NP-complete problem. It means that the time for solving this problem grows exponentially wrt to its size. In Chapter 3 we describe how efficient representations of qualitative systems of constraints, leading to effective algorithms, were used to get further insights into the model under study. We shall see that these algorithms are able to deal with large-scale networks.

2.2.2 Main concepts of the consistency analysis

As we saw in the previous example, the analysis of a regulatory network, by studying its qualitative system of constraints, is composed of specific elements. We will describe them in detail in the following paragraphs. The ability to extract these elements from a biological problematic will allow us to analyze large-scale regulatory networks wrt experimental measurements using this approach. As we will discuss in the next sections, several types of analyses can be proposed by specifying these elements differently.

I. Influence graph and experimental measurements. These objects correspond to the biological information we initially have of the organism under study. This information need to be represented qualitatively. For this reason we need to identify:

(1) the $\{+, -, \mathbf{?}\}$ signs of the edges in the graph, using the influence graph information, and

(2) the $\{+, -\}$ variations of the nodes of the graph, using the experimental measurements information.

In most of the cases we cannot precise this information thoroughly. Thus, we can only provide partial values for the variations of the nodes and/or signs of the edges. The non-observed signs (of edges and nodes) will correspond to the ***variables*** of the qualitative system of constraints.

In a first step of the analysis, either the signs of the edges or the signs of the nodes may be fixed. The further analysis of the qualitative system of constraints will provide new values for the non-observed signs.

II. Qualitative constraints. A qualitative constraint relates the qualitative information of the influence graph and experimental measurements. Specifically, it relates the variation of a network node with the influences it receives from its direct predecessors in the influence graph, as shown in Equation 2.1.

The way we define F_T in this relation depends, once more, on the biological information of the organism under study. In Section 2.1.4 we showed a generic consistency constraint composed by the GEN_T qualitative function.

As a general rule, it is difficult to determine the priority or weight of biological regulations. In these cases, imposing the generic qualitative constraint suits the best. However, in some cases it is possible to model F_T precisely by using biological information. Notice that if F_T is known precisely, the signs of the edges connecting T with its predecessors do not need to be specified.

Once the qualitative functions are chosen, the qualitative system of constraints for an influence graph can be generated automatically by the informatic tools proposed in Chapter 3.

III. Solving a qualitative system of constraints. The solution of a system of qualitative constraints contains interesting information concerning the confrontation with the biological data. The research of this solution outputs three main results that will be recurrently used during this thesis.

1. **Consistency:** as seen in Section 2.2.1, the solution of a system of qualitative constraints is reached when a $\{+, -\}$ valuation of all the variables of the system exists. This valuation must satisfy all of the system constraints. The consistency answer is given as a Boolean *true* or *false* value.

2. **Hard components - predictions:** when the system of constraint is consistent, at least one solution of the system exists. This means that the variables of our system are assigned either '+' or '−' values. Let us say, for example, that in all the proposed solutions the sign value of the molecule *LacI* is fixed to '+'. Then, *LacI* will be considered as a *hard component* of the system of constraints. Hard components are also called *predictions*, since these values correspond to the explanations of our initial qualitative observations.

3. **Inconsistent subgraph:** when the system of constraints is inconsistent we can find the minimal set of constraints responsible for the inconsistency. This set of constraints can correspond, for example, to a subgraph of the network in which the signs of the edges do not explain the variations of the nodes observed in the experimental measurements.

Notice that for finding the predictions of the system we need to analyze the solution of the system of constraints. However, for finding the inconsistent subgraph we need to analyze the system of constraints. For small and simple networks this reasoning can even appear natural. The difficulty is to detect these concepts when studying large-scale regulatory networks, as well as genome-wide measurements. In Chapter 3 we will discuss the considered informatic approaches to compute the consistency, predictions, and inconsistencies of the large-scale networks.

Elements contained in points **I** and **II** can be considered as input of our reasoning, while elements in **III** can be the output. The mapping of biological data into qualitative information as well as the constraints choice are the steps that have to be performed before the consistency analysis. While these steps may require biological expertise, obtaining the results shown in **III** only requires informatic skills.

2.2.3 Analyzing large-scale signed transcriptional networks

The analysis of the large-scale TRNs has as an objective to confront a large-scale known (signed) influence graph, issued from a TRN, wrt a single dataset containing genes differential expression between two conditions. Once this confrontation is made two possible alternatives appear: (1) we may predict variations of the network products when a network and a single dataset are consistent, or (2) we may generate inconsistent graphs, including the hypothesis of the inconsistency origins.

In what follows we will describe which type of biological data is required, which type of constraints we need to impose over this data, and the logical steps needed to achieve the expected results. The algorithms performing these steps will be detailed in Chapter 3.

2.2.3.1 Biological data input

The input data required for the analysis is a regulatory network and a single dataset. Regulatory networks should have a known topology and a partial labeling of their edges, that is, the sign and the direction of the interaction among two network products have to be known. The network can be composed of molecular elements such as genes, proteins, protein-complexes, and active proteins. The interactions among products represent *causal* relationships; a natural example of this type of interactions is a TF-gene relation. The edges of an influence graph are labeled qualitatively as $+$, $-$, and **?**, where $+$ represents a positive influence, $-$ a negative influence, and **?** a dual or complex regulation.

We also require a dataset of differential molecules expression issued from perturbation steady state experiments. This dataset must be composed of qualitative $\{+,-\}$ *concentration changes* of some of the molecules in the network. One type of concentration changes may be statistically significant mRNA-expression responses, described qualitatively as: $+$ up-regulation, $-$ down-regulation. We may also provide other reliable concentration changes issued from the literature or other types of stress perturbation experiments.

2.2.3.2 Mathematical constraints

The system of qualitative constraints for this problem is built by imposing the generic constraint $t \simeq GEN_T$ (equation 2.2) over each inner product T of the network, such that T does not self-regulate positively, that is:

$$t \simeq \sum_{j\in\{1,\cdots,p\}}^{\oplus} sign(S_j \rightarrow T) \otimes s_j,\ \forall S_j \neq T \tag{2.4}$$

In this equation, $sign(S_j \rightarrow T)$ is fixed to $\{+,-,?\}$ values obtained from the signed influence graph. The values for t (resp. s_j) will be *partially* fixed by the variations of

network products reported in the dataset. The network products that were not observed in the dataset will correspond to the variables of the system.

Notice that by identifying the variables in the system of constraints, we identify which type of predictions we may obtain. In the case our qualitative system of constraints is consistent, the predictions obtained will correspond to the non-observed network products. That is, we will predict the $\{+, -\}$ change in expression of some network nodes.

2.2.3.3 Steps of the analysis

In the following we list the steps of the consistency analysis of large-scale signed transcriptional networks:

1. Build a signed influence graph from the TRN.
2. Collect the variations of the nodes from an experimental dataset.
3. From elements in points 1 and 2 build the system of qualitative constraints using Equation 2.4.
4. Compute the consistency answer of the system of qualitative constraints:
   - If the system is consistent, generate predictions over the non-observed variations of the nodes.
   - If the system is inconsistent, detect the inconsistent graph.

The predictions generated in Step 4 can be afterwards validated with wet lab studies. Recall that not only genes but also proteins could compose the network. The computational predictions can be afterwards compared wrt RNA microarray measurements. In the case of contradictory results between the RNA measured expression and prediction of a gene, we could generate hypothesis on the post-transcriptional regulations targeting this gene under the studied experimental condition.

The inconsistencies generated in Step 4 can highlight the nodes and edges associated with a subnetwork. This could question: *(i)* the completeness of the network, *(ii)* the accuracy of an observed data point in the dataset, or *(iii)* the accuracy of the proposed edge labeling of the network.

In Fig. 2.4 we illustrate the flow-chart of the complete consistency check process for this analysis.

In order to analyze large-scale networks using this approach we require qualitative information on network regulations as well as on expression variations of the molecules. In Chapter 3 we describe how a large system of constraints can be computationally solved in a reasonable time. An efficient data structure together with well performing algorithms were included into a library named Bioquali; using this library we generated the program instructions used to compute the automatic steps shown in Fig. 2.4. This analysis was the most tested during this thesis. In Chapters 5 and 6 we will show the

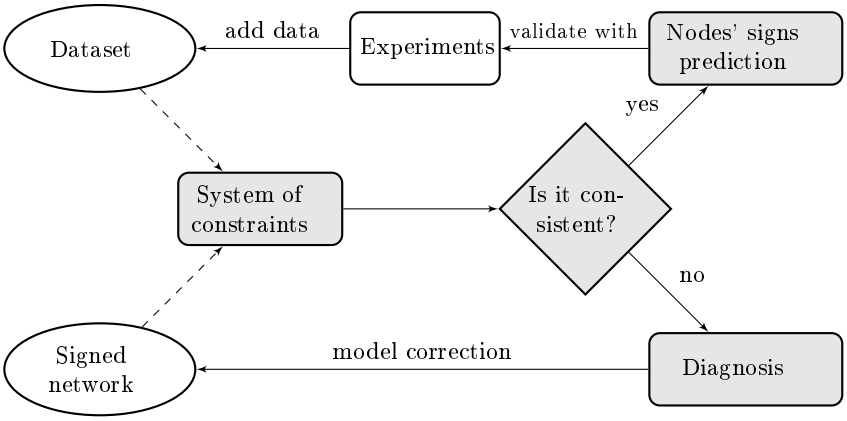


Figure 2.4: Consistency check process for a network modeled using only generic qualitative constraints. (1) We build a system of constraints from an influence graph (network) with a dataset of concentration changes, (2) we check the consistency of the system, and (3) if it is consistent and an initial dataset was provided, we may predict new concentration changes of the molecules in the network. These predictions can be compared with real measurements and question the original dataset and model. If it is not consistent, we report the inconsistent region in order to correct the network or initial dataset. Note, that for the sake of figure clarity the arrow from *Diagnosis* to *Dataset* is not shown. The shaded blocks represent the automatic generated outputs from our analysis.

results obtained for the large-scale network of *E. coli* and for the signaling network of the EWS-FLI1 human oncogene. In addition to the possibility of computing this analysis automatically using the Bioquali library, we also developed bioinformatic tools that accessed this reasoning via a Website, a Cytoscape plugin, and a Web service (see Chapter 4).

2.3 Analyzing networks including post-translational regulations: looking for the origin of causes

In Section 2.1.4 we modeled the total contribution of the influences targeting a node T in the network, using the generic function GEN_T. In this section we propose new ways to model F_T that precise the relation of T with its predecessors. The precise modeling of F_T searches to expand the range of regulatory networks that can be analyzed with our approach. In particular, we search to analyze regulatory interactions representing post-translational effects, which appear commonly in signaling networks.

In addition, by manipulating the regulations targeting a node in the network, we can investigate the origin of the causes of its observed change in experimental measurements. We will detail these ideas in the following paragraphs.

We begin by presenting which new ways of modeling F_T were proposed and their

justification, following by how to investigate the causes of an observed change of a network product.

2.3.1 Protein complex constraint

In this section we propose a qualitative constraint used to describe the qualitative variation of a protein complex. It describes the variation of a protein complex concentration by taking into account the RNA concentration of its subunits. This rule can be intuitively expressed as *the weakest takes it all*, for it states that the complex concentration at steady state follows the concentration of the subunit with the lowest concentration. The same rule was used elsewhere [RZL07] to obtain dominant simplifications (called min-funnel) of signal transduction models.

2.3.1.1 Mathematical justification

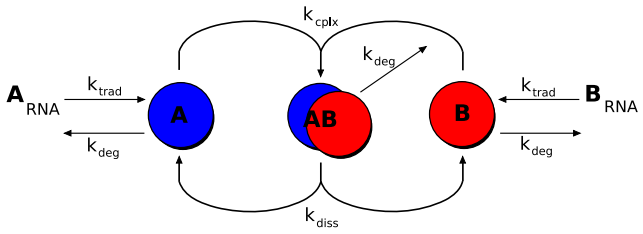


Figure 2.5: Representation of the different processes affecting the formation of an heterodimeric protein complex.

We assume that A and B are the two subunits of the protein complex AB. The components of our model are: A, B, AB, and the RNA precursors of A and B, namely A_{RNA} and B_{RNA}. This model is depicted in Fig. 2.5. We use a mass action kinetic law to describe the formation of the complex. We thus introduce rates associated to the different reactions:

- k_{deg} is the constitutive degradation rate of proteins A, B, and the AB complex;
- k_{trad} is the translational production rate of proteins A and B;
- k_{cplx} is the rate of formation of the complex AB; and
- k_{diss} is the rate of dissociation of the complex AB.

Proposition 2.3.1 (the weakest takes it all). *Assume that*

$$\frac{k_{trad}k_{cplx}}{k_{deg}(k_{deg}+k_{diss})}|[B_{RNA}]_e - [A_{RNA}]_e| >> 1.$$

Then the concentration of the complex $[AB]$ at steady state is proportional to the smallest of the two concentrations of the RNAs:

$$[AB]_e \approx \frac{k_{trad}}{k_{deg}} \min([A_{RNA}]_e, [B_{RNA}]_e).$$

We now include the system depicted in Fig. 2.5 in a larger system that can be perturbed, implying that the RNA concentrations can slowly change to induce a shift of equilibrium of the small system. Using the relations described above it becomes possible to predict the qualitative behavior of the system.

Proposition 2.3.2. *Assume that the system depicted in Fig. 2.5 is a part of a larger system that is slowly perturbed from a steady state $e1$ to a steady state $e2$. Assume that the concentration of B_{RNA} is always smaller that the concentration of A_{RNA} during the experimentation.*

Then the variation of the complex concentration AB between the two steady states have the same sign as the variation of the RNA of the protein with the lowest quantity:

$$[A_{RNA}] \geq [B_{RNA}] \implies sign([AB]_{e2} - [AB]_{e1}) = sign([B_{RNA}]_{e2} - [B_{RNA}]_{e1}).$$

The proofs of the propositions mentioned in this section are given in the supplementary material of [GGRS09], at `http://www.irisa.fr/symbiose/networks/consistency.html`.

Qualitative constraint for a protein complex

We introduce now the qualitative constraint over the variation sign of a node T, when T is a protein-complex formed by proteins S_1 and S_2:

$$t \simeq minimum(S_1, S_2) \tag{2.5}$$

where t represents the $\{+,-\}$ variation of molecule T, and $minimum(S_1, S_2)$ outputs s_1 (resp. s_2), the $\{+,-\}$ variation of molecule S_1 (resp. S_2), depending on:

$$minimum(S_1, S_2) = \begin{cases} s_1 & \text{if } [S_2^{eq2}] > [S_1^{eq2}] \textbf{ and} \\ & S_1 \text{ had the smallest concentration during the experiment.} \\ s_2 & \text{if } [S_2^{eq2}] < [S_1^{eq2}] \textbf{ and} \\ & S_2 \text{ had the smallest concentration during the experiment.} \end{cases}$$

Notice that in order to apply this constraint, we need to have *quantitative* information about the molecules forming the complex.

2.3.1.2 Applying the protein complex constraint

We applied the protein complex constraint to the IHF protein complex, an important *E. coli* transcription factor. The integration host factor (IHF) is an heterodimeric protein complex formed by two polypeptides α and β, transcribed by genes *ihfA* and *ihfB*. The experimentation we considered consists of a growth shift from exponential phase to stationary phase of *E. coli* cells. At complex-formation timescales both can be considered to be steady states of the system.

The rate condition given in Prop. 2.3.1 is satisfied since: *(i)* IHF is a metabolically stable dimer [GN93], meaning that the complex formation rate k_{cplx} is high, and *(ii)* at least one of the proteins IhfA, IhfB, or the IHF complex is present in the system, as changes in their targets are observed [AHL$^+$03].

The variation condition given in Prop. 2.3.2 is satisfied because the RNA concentration of *ihfB* during the exponential growth phase (condition $e1$) is observed in a lower quantity than the RNA concentration of *ihfA* (Affymetrix dataset from [AHL$^+$03]). Once we move to the stationary phase (condition $e2$), the RNA concentration of *ihfB* decreases, whereas the RNA concentration of *ihfA* increases (see Fig. 2.6).

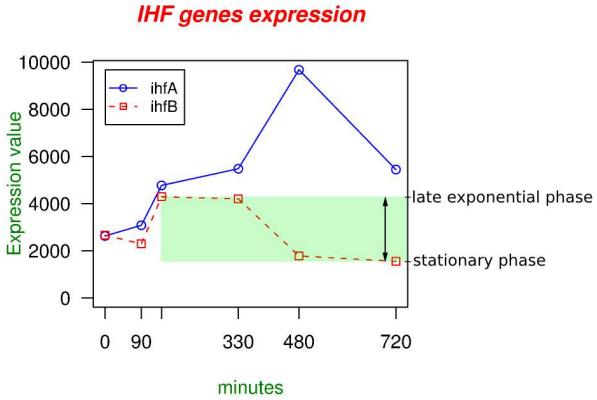


Figure 2.6: Expression levels of the genes coding for the protein subunits of the IHF complex extracted from [AHL$^+$03]. The green box refers to the change in RNA level of the limiting subunit (IhfB) during the transition from exponential phase to stationary phase.

Prop. 2.3.2 then applies. We deduce that the concentration of the IHF protein complex decreases when comparing cells in a stationary phase with cells in a late exponential phase. Notice that this result holds independently from the behavior of *ihfA*, which is far from being regular; the only information we use from *ihfA* is that its RNA concentration is always larger than the RNA concentration of *ihfB*. We can, therefore, state the following constraint over IHF:

$$sign(IHF) \simeq sign(ihfB)$$

and by replacing the known concentration change of *ihfB* we deduce that:

$$[IHF]_{Log-phase} > [IHF]_{Stat-phase}$$

that is, '$sign(IHF) = -$'. In Chapter 5 we will see how the application of this new constraint influenced our prediction results when analyzing the transcriptional *E.coli* network.

2.3.2 Specific qualitative constraints

In the previous subsection we saw how to precise the modeling of F_T by using quantitative information of the system. In this section we illustrate some biological examples where we propose new ways to model F_T. These regulatory functions were proposed using the qualitative information as well as the biological literature of the system under study.

Adding specificity to the generic qualitative constraint has an interest regarding the number of known behaviors of the system. In Chapter 6 we show how, by adding more precise modeling into the system, we can affect the number of total predictions. We were also able to reduce the probability of generating the same predictions with random data.

2.3.2.1 Boolean sign operators

As a general rule, it is difficult to determine the priority or weight of biological regulations. However, in some cases it is possible to model F_T precisely. On that account, we introduced the regulatory functions AND and OR, that may be used to represent accurately a regulation in the network. These functions are defined as:

$$AND_T(s_1, s_2, \cdots, s_p) = s_1 \wedge s_2 \wedge \ldots \wedge s_p \quad (2.6)$$

$$OR_T(s_1, s_2, \cdots, s_p) = s_1 \vee s_2 \vee \ldots \vee s_p \quad (2.7)$$

where $\{s_1, \ldots, s_p\}$ represent the variations of the p predecessors of T, and the $\wedge$ and $\vee$ sign operators are described in Table 2.4. The $\neg$ operator was also added to output the opposite sign of a variation, *i.e.* $\neg t = t \otimes -$.

Table 2.4: $\wedge$ and $\vee$ sign operators.

$\wedge$	$+$	$-$	**?**
$+$	$+$	$-$	**?**
$-$	$-$	$-$	$-$
?	**?**	$-$	**?**

$\vee$	$+$	$-$	**?**
$+$	$+$	$+$	$+$
$-$	$+$	$-$	**?**
?	$+$	**?**	**?**

2.3.2.2 Examples of specific qualitative functions

The sign operators described in Table 2.4 are used to represent more complex regulatory phenomena. For example, in the study of the signaling network of the EWS-FLI1 human

oncogene, we detected three mechanisms that deserved to be modeled in a different way than by using the generic qualitative function GEN:

Strong inhibitor The RB protein family are canonical members of tumor-suppressors. They act as default inhibitors of E2F protein family, which are transcription factors implied in cell cycle progression. According to [DeG02], RB proteins act by sequestrating E2F. When RB proteins are phosphorylated, E2F are released. The releasing triggers cell division, in particular transition to S phase. An illustration of these regulations is given in Fig. 2.7A, where the valuations of E2F, RB, and cell_cycle_S are consistent with respect to the literature. Consequently, the inhibition of E2F by RB was modeled using a *Strong-inhibitor* function, Equation 2.8, adapted to all default inhibitors.

Complex inactivation Most of the protein complex formation in our networks are modeled using the GEN function. However, some proteins can hamper the protein complex formation. As discussed in [OS02], this is the case for the complex CCNE-CDK2: WEE1 phosphorylates CDK2 on tyrosine and threonine residues, causing the complex inactivation. In Fig. 2.7B we illustrate this case and we model it using the *Complex-inactivation* function shown in Equation 2.9.

Complex inactivation-reactivation In some cases a protein complex may be under an inactivation influence that is itself inhibited. For example, as shown in [PRKG+99], CCNA forms a complex with CDK2, which is inhibited by cycline-dependent-kinase-inhibitor CDKN1A. This inhibition is reverted by CCND-CDK4-a, the active complex formed by Cycline D and cycline-dependent-kinase CDK4. This situation is illustrated In Fig. 2.7C. We model it using the *Complex-inactivation-reactivation* function shown in Equation 2.10.

The three qualitative functions added to model these situations are listed below:

$$F_{cell\_cycle\_S} = e2f \wedge \neg rb1 \quad (2.8)$$

$$F_{CCNE\_CDK2} = \neg wee1 \wedge (ccne \oplus cdk2) \quad (2.9)$$

$$F_{CCNA\_CDK2} = (ccna \oplus cdk2) \wedge (\neg cdkn1a \vee ccnd\_cdk4\_a) \quad (2.10)$$

In Chapter 6 we will describe in detail the results obtained by using these new qualitative functions in the system of constraints of the EWS-FLI1 signaling network.

2.3.3 Analyzing regulatory networks including specific rules

The analysis steps of a system of qualitative constraints that includes precise modeling of the F_T function are similar to the analysis of transcriptional networks (Section 2.2.3). However, two additional steps were added:

1. Build a signed influence graph from the regulatory network.
2. Collect the variations of the nodes from an experimental dataset.

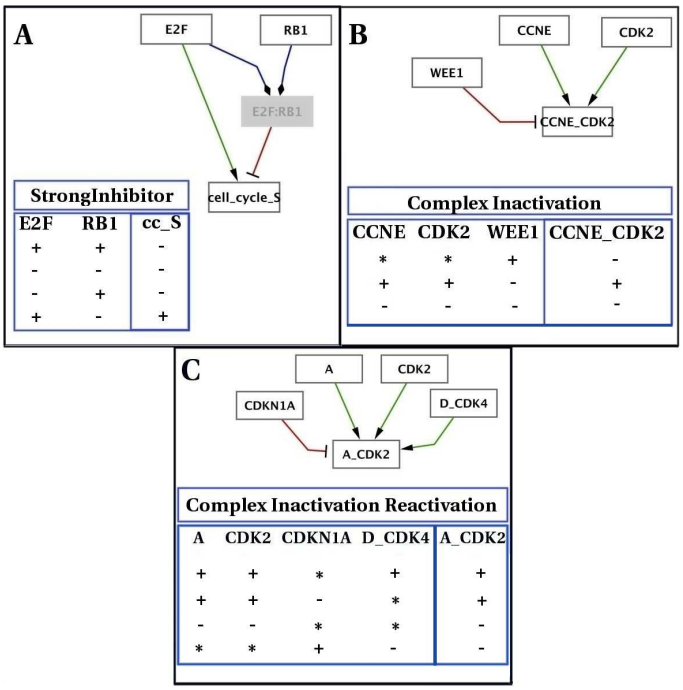


Figure 2.7: Regulations from the EWS-FLI1 network modeled using complex functions. In each table we show all the possible $\{+,-\}$ variations of the predecessors of a node, so that the node will be predicted to a $\{+,-\}$ value. The * symbol refers to either '+' or '−' variation.

3. For the network nodes where biological knowledge precise the regulations they receive, add new specific rules to model F_T. For the others, use the generic qualitative rule GEN_T.

4. From elements in points 1, 2, and 3 build the system of constraints.

5. Compute the consistency answer of the qualitative system of constraints:

   - If the system is consistent, generate predictions over the non-observed variations of the nodes.
   - If the system is inconsistent, detect the inconsistent graph.

6. If the system was consistent, analyze once more the set of predictions in order to explain the causes of some observed products.

In step 6 the qualitative functions AND and OR (see Equations 2.6 and 2.7) are used to provide automatic explanations of the predicted sign of a network component. Thus, we can answer to the questions: Which are the molecules responsible of a change in the network? Which molecules need to be switched on/off to obtain a specific change? The algorithms used to answer these questions, as well as the results obtained for the EWS-FLI1 signaling model, will be given in chapters 3 and 6 respectively.

In Fig. 2.8 we provide a flow-chart including the new inputs and outputs of this analysis. We can see that the flow-chart proposed previously in Fig. 2.4 was extended. Two functionalities were added: *(i)* the possibility to add new qualitative regulatory functions F_T, and *(ii)* by using the AND and OR qualitative functions, we can reason over the causes that explain some observations in the dataset. The functionality *(ii)* takes advantage of the larger number of predictions generated when using a more specific modeling.

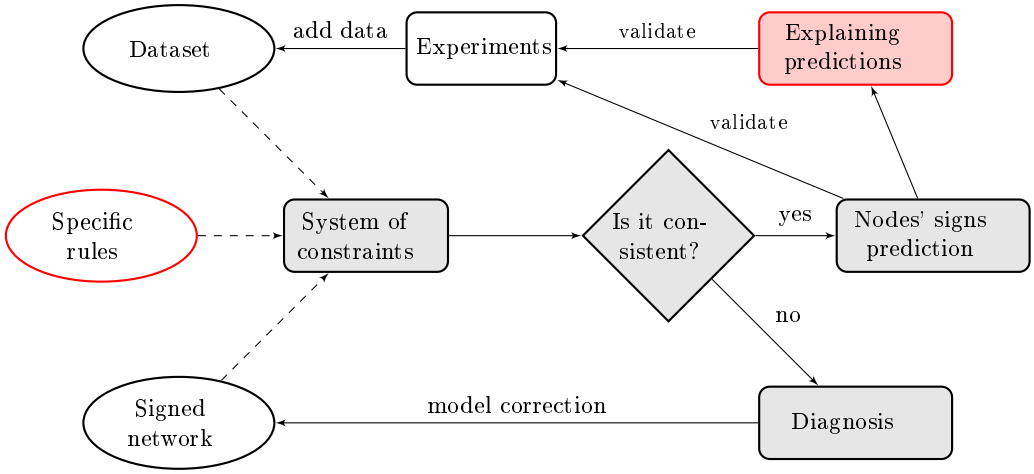


Figure 2.8: Consistency check process for a signaling network modeled using more specific rules of regulations in addition to the generic rule. The steps are the same as those presented in Fig. 2.4. In addition, we are able to add new rules describing the regulations targeting a node in the network in more detail. Also, we are able to perform a post-analysis of the predictions obtained in order to search for the origin of a network product variation, observed in the dataset. The red blocks are the new functionalities added wrt the consistency process based only on the generic qualitative constraints. Again, for the sake of figure clarity the arrow from *Diagnosis* to *Dataset* is not shown.

2.4 Analyzing unsigned regulatory networks

We are interested now on determining the regulatory role of a TF on its target genes given a set of expression profiles. In the previous sections we approached the analysis of a regulatory network in terms of finding the variations of network molecules between two steady states. The variables of the system of constraints were the sign variations of the non-observed nodes. In this new problem, the variables of our system of constraints will be the signs of the edges.

2.4.1 Biological data input

To solve this problem we require a regulatory network with a known topology, however, the signs of its edges may stay unknown. Its influence graph is either given by a regula-

tory network to be validated, or built from chIP-chip data and TF binding site search in promoter sequences. Thus, as soon as a TF j binds to the promoter sequence of gene i, j is assumed to regulate i. This is represented by an arrow $j \rightarrow i$ in the influence graph.

In addition, we require ***multiple*** expression profiles issued from different perturbation experiments. Previously, we considered only *stress* perturbation experiments. In this new problem, in addition to stress perturbation experiments we consider *genetic* perturbation experiments, where a gene of the cell is either knocked-out or over-expressed, and where perturbed cells are compared to the reference. Thus, expression profiles provide the sign of variation of the gene expression for a set of r steady-state perturbation, mutant, or over-expression experiments.

2.4.2 Mathematical constraints

The system of qualitative constraints for this problem is built by imposing the generic constraint $t \simeq GEN_T$ (equation 2.2) over each inner product T of the network, such that T does not self-regulate positively, that is:

$$t \simeq \sum_{j \in \{1,\cdots,p\}}^{\oplus} sign(S_j \rightarrow T) \otimes s_j, \ \forall S_j \neq T \tag{2.11}$$

When the experiment is a genetic perturbation, the same equation holds for every node that was not genetically perturbed during the experiment and such that all its predecessors were not genetically perturbed. If the predecessor S_m of T was knocked-out, the equation becomes:

$$t \simeq -sign(S_m \rightarrow T) \oplus \sum_{j \in \{1,\cdots,p\}, j \neq m}^{\oplus} sign(S_j \rightarrow T) \otimes s_j \tag{2.12}$$

The same holds with $+sign(S_m \rightarrow T)$ when the predecessor S_m was over-expressed. There is no equation for the genetically perturbed node.

To infer the regulatory role of the TFs we assume that the regulatory role of a TF j on a gene i (as inducer or repressor) is represented by the variable S_{ji}, which is constrained by Equations 2.11 and 2.12. In the following, X_i^k will stand for the sign of the observed variation of gene i in experiment k, and r for the total number of experiments we have. Our inference problem can now be stated as finding values in $\{+,-\}$ for S_{ji}, subject to the constraints:

for all $(1 \leq i \leq n)$, $(1 \leq k \leq r)$, s.t. i not genetically perturbed in the k^{th}experiment

$$\begin{cases} X_i^k \simeq \sum_{j \in pred(i)} S_{ji} \otimes X_j^k, \text{ if all predecessors } j \text{ are not genetically perturbed} \\ X_i^k \simeq -S_{mi} \oplus \sum_{j \in pred(i), j \neq m} S_{ji} \otimes X_j^k, \text{ if } m \text{ is knocked-out} \\ X_i^k \simeq S_{mi} \oplus \sum_{j \in pred(i), j \neq m} S_{ji} \otimes X_j^k, \text{ if } m \text{ is over-expressed.} \end{cases} \tag{2.13}$$

The variables of this system of constraints are the signs of the edges S_{ji}. The signs of the nodes X_i^k are given by the datasets. Therefore, if the system of constraints is consistent, the predictions will be associated to the signs of the edges, that is, the role of the TFs over their target genes.

2.4.3 Steps of the analysis

The TF role inference problem has more constraints than the analysis proposed in Section 2.2.3. Thus, it requires a longer computation time (see Chapter 3). The steps of this analysis are:

1. Build an unsigned influence graph from an unsigned regulatory network.
2. Collect the variations of the nodes from many experimental datasets.
3. Apply the constraints 2.13 to the gathered qualitative biological data.
4. Compute the consistency answer of the qualitative systems of constraints:
    - If the system is consistent, generate predictions over the non-observed signs of the network edges.
    - If the system is inconsistent, detect the inconsistent graph.

The inferred TF roles (signs of the edges) on Step 4 could also be validated with biological experiments; this validation could afterwards help to update the model of regulations. The inconsistencies reported on Step 4 may highlight: *(i)* two different roles for a TF under different experimental conditions, *(ii)* missing regulations in the model, or *(iii)* errors in data points of certain datasets.

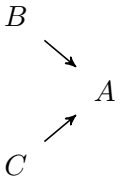
Let us illustrate this analysis on a very simple (yet informative) example. Suppose that we have a system of three genes A, B, and C, where B and C influence A. Let us say that for this influence graph we obtained six experiments, and that in each of them the variation of all products in the graph was observed (Table 2.5). Using some or all of the experiments provided will lead us to different qualitative constraint systems, as shown in Table 2.6, hence to different inference results.

In Chapter 3 we detail how the Bioquali library was used to obtain automatically this analysis. We applied this process on the transcriptional regulatory networks of *E. coli* and *S. cerevisiae*; in Chapters 5 and 6 we detail our results.

2.5 Synthesis

In Table 2.7 we summarize the different types of modeling proposed to approach different biological questions.

Table 2.5: Influence graph of three genes A, B, and C, where their change in expression was observed by six stress perturbation experiments.



Stress perturbation expression profile	X_A	X_B	X_C
e_1	+	+	+
e_2	+	+	−
e_3	−	+	−
e_4	−	−	−
e_5	−	−	+
e_6	+	−	+

Table 2.6: TF role inference process. The variables of the system of constraints are the signs of the network edges. Variations of the species in the graph are obtained from six experiments (Table 2.5). Using different profiles we can infer different roles of regulation. Using $\{e_1, e_2, e_3\}$, we obtain a system with three qualitative constrains; not all valuations of variables S_{BA} and S_{CA} satisfy this system according to the sign algebra rules. When we obtain unique values for these variables in the solution of the system, we consider them predicted. *Exp.* refers to experiments, and *Const.* to constraints.

Exp. used	Const. for	Replacing values from experiments	Consistent solutions (S_{BA}, S_{CA})	Prediction
$\{e_1\}$	X_A^1	$(+) \simeq S_{BA} \otimes (+) \oplus S_{CA} \otimes (+)$	(+,+) (+,−) (−,+)	$\emptyset$
$\{e_1, e_2\}$	X_A^1 X_A^2	$(+) \simeq S_{BA} \otimes (+) \oplus S_{CA} \otimes (+)$ $(+) \simeq S_{BA} \otimes (+) \oplus S_{CA} \otimes (-)$	(+,+) (+,−)	$\{S_{BA} = +\}$
$\{e_1, e_2, e_3\}$	X_A^1 X_A^2 X_A^3	$(+) \simeq S_{BA} \otimes (+) \oplus S_{CA} \otimes (+)$ $(+) \simeq S_{BA} \otimes (+) \oplus S_{CA} \otimes (-)$ $(-) \simeq S_{BA} \otimes (+) \oplus S_{CA} \otimes (-)$	(+,+)	$\{S_{BA} = +, S_{CA} = +\}$

Table 2.7: Summary to the modeling applied to answer to different biological questions. The *Network* column refers to the type of regulatory network on which the analysis was applied. The *Constraints* column refers to the type of constraints used to build the qualitative system of constraints (QSC). *Process* refers to the analyses performed over the QSC in order to answer the biological questions.

Biological question	**Network**	**Constraints**	**Process**
Does it exist at least one explanation of the variations obtained from the influences that a molecule receives?	Transcriptional and signaling networks	• GEN_T (Eq. 2.2) • precise F_T (Eq. 2.1)	Asking whether the QSC is consistent. Consistency-check processes (Figs. 2.4, 2.8).
Which variations of the molecules can be deduced from partial observations and from the network structure?	Transcriptional and signaling networks	• GEN_T (Eq. 2.2) • precise F_T (Eq. 2.1)	Predictions of a consistent QSC. Consistency-check processes (Figs. 2.4, 2.8)
Which network interactions and/or dataset observations must be modified to explain the variations of the molecules?	Transcriptional and signaling networks	• GEN_T (Eq. 2.2) • precise F_T (Eq. 2.1)	Inconsistent constraints of a QSC. Consistency-check processes (Figs. 2.4, 2.8)
Which regulatory roles must be used to explain multiple stresses observations?	Unsigned transcriptional networks	• TF inference constraints (Eq. 2.13)	Predictions of the QSC. TF role inference analysis (Section 2.4)
Which molecules can be responsible for a network product $\{+,-\}$ observed change?	Signaling networks	• GEN_T (Eq. 2.2) • precise F_T (Eq. 2.1) • AND_T (Eq. 2.6) • OR_T (Eq. 2.7)	Post-analysis of the QSC predictions. Consistency-check process (Fig. 2.8)

Chapter 3

Algorithms to analyze large-scale networks

In this chapter we show on which informatic solutions lied many of the results of the research presented in this thesis. Two informatic approaches were proposed to deal with the consistency analysis of the large-scale regulatory networks. The first, based on ternary decision diagrams (TDDs), evolved through time and it was adapted to solve new questions and problems we were confronted with. The second, uses Answer Set Programming (ASP) to approach the same problematic. The objective of our research was to test, correct, and use these efficient implementations in order to analyze large-scale and complex biological networks. It is not our intention to go in the deep details of the implementation of these approaches, but to present the programs we proposed (based on these approaches), to answer many questions concerning the confrontation of qualitative large-scale biological data.

3.1 Using decision diagrams to solve qualitative constraints

3.1.1 Main concepts

In Chapter 2 we defined the qualitative system of constraints as the mathematical formalization of a regulatory network. This system of constraints was composed of variables that could take three values: $\{+, -, \mathbf{?}\}$. Computing the satisfiability of a large system of qualitative constraints is an NP-complete problem for even linear qualitative systems [Dor88] and classic methods of resolution do not allow to solve this kind of problems [TMD03].

In [MBLBLG00] polynomial functions on $\mathbb{Z}/3\mathbb{Z}$ were represented using a ternary decision diagram (TDD) data structure, which is an extension of the BDDs (binary decision diagrams) used in [Bry86] to code polynomial functions with hundred of variables. TDDs were proposed and applied for the verification of signal processing programs using a program named SIGALI.

The interest of TDDs is that they can represent polynomial functions on $\mathbb{Z}/3\mathbb{Z}$ in

a compact way. The starting point of this idea is the Shanon decomposition for logical functions, given by the following expression:

$$p(X_1, X) = (1 - X_1^2)p_{[X_1=0]}(X) + X_1(-X_1 - X_1^2)p_{[X_1=1]}(X) - X_1(X_1 - X_1^2)p_{[X_1=-1]}(X) \quad (3.1)$$

where $p(X)$ is a polynomial function with n variables. Hence, this polynomial function can be recursively represented by a tree: the variable X_1 is the root and has three children. Each children is obtained by instantiating X_1 to 0, 1, or -1 in $p(X_1, X)$. This representation is exponential in the number of variables (3^n). In [Bry86] it was observed that many of the sub-trees of this initial tree were identical, and thus its representation can be simplified. If the identical sub-trees are represented only once, then the tree representation is transformed into a direct acyclic graph forming a TDD. This TDD will represent the same polynomial function, but in a compact manner (see Fig. 3.1). Its size will depend on the order of the variables.

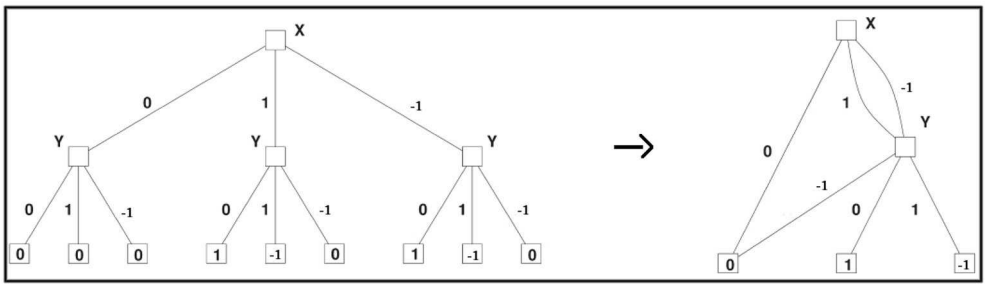


Figure 3.1: Tree representation (left) of the polynomial function $X^2(Y+1)$. It has two identical sub-trees, thus this representation can be reduced to a TDD structure (right). Extracted and adapted from [VBSR05].

Following this direction, in order to implement efficient computations on sign algebra, the authors of [LBP07] coded the sign variables and algebra into a $\mathbb{Z}/3\mathbb{Z}$ field as $\{1, -1, 0\}$ (see Table 3.1), and added several algorithms to SIGALI to answer questions of biological interest [VBSR05]. The first approach used to solve a qualitative system of constraints was detailed in [LBP07], [VBSR05], and [Veb07]. It can be summarized in the following steps:

1. *Reduce the influence graph so that it preserves the system satisfiability.* This step is achieved by iteratively removing the nodes of the influence graph that are not observed and have no successors in the graph. The result of this procedure is a subgraph such that any node is either on a cycle, or has a cycle downstream. This subgraph will be represented afterwards as a qualitative system of constraints. In [VBSR05] it was shown that the global consistency of the whole network is equivalent to the consistency of its reduced subgraph. The reduced subgraph is also called the core of the network.

2. *Transform qualitative constraints in sign algebra into a polynomial function with multiple variables to be solved over the finite field* $\mathbb{Z}/3\mathbb{Z}$. A natural mapping from

the sign algebra to this field allows us to interpret the consistency relation $\simeq$ as a simple $\mathbb{Z}/3\mathbb{Z}$ relation (see Table 3.1). In this field, every constraint appears to be a polynomial function, and the zeros of a system of constraints are, equivalently, the zeros of a unique function.

3. *Efficiently represent the qualitative system of constraints.* As we saw at the beginning of this section, the TDDs offer an optimal structure to represent polynomial functions in a compact way. Once the system of qualitative constraints was reduced, all the qualitative constraints, written as $\mathbb{Z}/3\mathbb{Z}$ relations, are represented using a TDD. In this decision diagram, each node represents a variable of the system of constraints, and the arrows outputting from a node correspond to the two possible values $\{+, -\}$. These values were chosen because a variable can take them in order to be a solution of the system. The terminal nodes in the decision diagram take the values 0 or 1. If the path describing an instantiation of the variables ends in 0, then the system of constraints is consistent with the given instantiation of the variables. If the path ends in 1, the system of constraints is inconsistent. Notice that after building the TDD describing a qualitative system of constraints, computing its satisfiability or consistency is straightforward.

4. *Efficient analysis.* Efficient algorithms that cover the decision diagram were developed in order to find the predictions of a consistent system of constraints, as well as to perform diagnosis in case of inconsistency.

Table 3.1: This table specifies how the sign algebra in $\{+, -, \mathbf{?}\}$ was mapped into the Galois field $\mathbb{Z}/3\mathbb{Z}$. Extracted from [VBSR05].

Sign algebra (e)		$\mathbb{Z}/3\mathbb{Z}$ ($\overline{e}$)	Sign algebra		$\mathbb{Z}/3\mathbb{Z}$
$+$	$\rightarrow$	1	$e_1 \oplus e_2$	$\rightarrow$	$-\overline{e_1}.\overline{e_2}.(\overline{e_1} + \overline{e_2})$
$-$	$\rightarrow$	-1	$e_1 \otimes e_2$	$\rightarrow$	$\overline{e_1}.\overline{e_2}$
$\mathbf{?}$	$\rightarrow$	0	$e_1 \simeq e_2$	$\rightarrow$	$\overline{e_1}.\overline{e_2}.(\overline{e_1} - \overline{e_2}) = 0$

This first approach, based on SIGALI, was able to map a regulatory network and a expression dataset file into a qualitative system of constraints, and then compute the *Yes* or *No* answer concerning the consistency of the system. Also, it could compute the predictions of the system in the case of consistency. The inconsistency diagnosis, however, did not handle a system with a hundred variables [Guz06]. In [VBSR05] they applied this informatic approach to a small scale model related to lipid metabolism (14 nodes). Afterwards, we applied this approach at large-scale considering the *E. coli* transcriptional network (1529 nodes, 3883 edges), extracted from RegulonDB on March 2006. We reported an inconsistency and, after corrections, a set of predictions (the details of these results are presented in Chapter 5). However, the diagnostic step was not an easy task, it was not totally automatized and resulted from parsing the file describing the system of constraints mapped into $\mathbb{Z}/3\mathbb{Z}$. In addition, SIGALI was not supporting simple programming functions as to iterate over an instruction. Accessing the algorithms of SIGALI was not an easy task for a prospective user.

To solve this problem, in [LBP07] the same algorithms were implemented as a library written in C programming language. Python bindings were also available providing easy access to these algorithms, and a Python package named Bioquali was proposed. This package was composed of Python modules and a module (pyquali) dedicated for intensive computation in qualitative algebra.

3.1.1.1 Functions to analyze signed TRNs

The methods of Bioquali aimed to confront a signed large-scale transcriptional network wrt one experimental dataset, were corrected and adapted to our problems. Before describing them let us introduce the following notation to represent a network and dataset:

- Let us denote $\mathcal{N} = (V, E, \sigma)$, the oriented and signed network to be analyzed, where V is the set of nodes, E the set of edges, and $\sigma : E \rightarrow \{+, -\}$ a labeling of the edges in the network.
- Let $\mu = \{(n, s) \mid n \in V, s \in \{+, -\}\}$ represent the dataset of partial significant variations of the nodes in $\mathcal{N}$.

Based on this notation the most important functions of the Bioquali library were:

1. BQ.core($\mathcal{N}$,μ) returns a subgraph of $\mathcal{N}$ that does not include non-observed nodes without predecessors.
2. BQ.consistency($\mathcal{N}$,μ) returns **true** if the network is consistent with the dataset, or **false** if not.
3. BQ.predictions($\mathcal{N}$,μ)$=$ $\{(n, s) \mid n \in V \setminus dom(\mu), s \in \{+, -\}\}$, is the set of predictions generated when $\mathcal{N}$ is consistent with μ.
4. BQ.inconsistentSubgraph($\mathcal{N}$,μ) outputs a minimal inconsistent subgraph $\mathcal{I} = (V_I, E_I, \sigma_I)$, where $V_I \subset V$, $E_I \subset E$, and $V_I \cap dom(\mu) \neq \phi$. Its nodes and edges explain the origin of the inconsistency.

The Python module that contains the four functions described above is called **Igraph**. We used these methods to construct programs that analyze signed or unsigned regulatory networks wrt to one or multiple gene expression profiles.

3.1.1.2 Functions to analyze unsigned TRNs

In order to analyze unsigned transcriptional networks, we added to the Bioquali package two new functions that build the system of constraints and reason over it when the variables of the system corresponded to the sign of the edges in the network. These functions used the following input information:

- An unsigned network, denoted as $\mathcal{N} = (V, E, \sigma)$, where V is the set of nodes, E the set of edges, and $\sigma : E \rightarrow \{X\}$ is not known.

- Multiple datasets, denoted as $M = \{\mu_1, \mu_2, \ldots, \mu_n\}$; where $\mu_k \in M$ is of the form $\mu_k = \{(n, s) \mid n \in V, s \in \{+, -\}\}$ and will contain partial observations of the variations of the network product in the k^{th} experiment.

Based on this notation, we added two functions to the Bioquali library in order to reason over this data:

- BQ.edgeConsistency($\mathcal{N}$,M) returns **true** if the network is consistent with all the datasets in M, or **false** if not.

- BQ.edgePredictions($\mathcal{N}$,M)$=\ \{(a, b, s) \mid (a, b) \in E,\ s \in \{+, -\}\}$, is the set of predictions generated when $\mathcal{N}$ is consistent with all $\mu_k \in M$.

3.1.1.3 Using dependency graphs to analyze signaling networks

The Igraph Python module was performing well for transcriptional networks that had a core of less than 100 nodes. However, when dealing with more complex types of networks, such as signaling networks, the computational time of the consistency analysis was unbearable. Therefore, a new strategy to compute the satisfiability of a system of constrains was proposed in [Bor09], in which a family of decision algorithms was presented taking advantage of the few variables shared among qualitative constraints. In this strategy the resolution of the set of constraints is based on the construction of a dependency graph as well as of a covering tree for it. Thus, the TDD structure is only used to represent a single constraint, *i.e.* the one relating a node with its direct predecessors in the network. The dependency graph is a bipartite graph which nodes represent variables and constraints; there is an edge between a variable x and a constraint f if x is a variable of f. The covering tree provides an order that combines conjunction operations and variable elimination, minimizing the complexity of each step. Using this optimal order, we were able to determine the consistency, diagnose the inconsistent constraints, and compute the predictions of larger and more complex systems in a reasonable time. The Python module containing the dependency graph representation is called **Dgraph**, and it can be found in the Bioquali Python package at `http://genoweb1.irisa.fr/Serveur-GPO/outils/help/datafiles_test/BioPyquali.tgz`.

Moreover, in order to analyze the precise rules of regulations present in signaling networks (see Section 2.3), the Bioquali library was extended with a better parser that allows to handle a wider range of influence types. Previously, there were only three possible influence types: $+$, $-$, and **?**. In the newest version of the Bioquali package it is possible to represent regulations with the following syntax: "`STRONG-INHIBITOR(A,B) -> C`", which may mean that C has two regulators, and that under some $\{+, -\}$ variations of A or B we can define a priority rule which tell us which one regulates C. The graph object representing the influence graph is built from this syntax, and a system of constraints is generated and analyzed in the same way as in the case of transcriptional networks.

3.1.2 Programs to study the consistency of large-scale networks

3.1.2.1 Consistency analysis of signed TRNs

Using the Bioquali functionalities described in Section 3.1.1.1 we designed the classic program shown in Algorithm 1 to analyze the consistency of a regulatory network wrt an experimental dataset as illustrated in the flow-chart of Fig. 2.4. We used this program to generate the results presented in [GGRS09], in which we *(i)* computed the consistency and *(ii)* predicted the qualitative variation of several network products for a larger version of the *E. coli* network (1915 nodes and 5140 edges; version of 2008 in RegulonDB) wrt to a dataset composed of 45 observations. The computation time of this analysis in case of consistency (resp. inconsistency) was of 10.8 s[1] (resp. 6.6 s). The details of our results are given in Chapter 5.

Algorithm 1 Consistency check process

Require: $\mathcal{N} = (V, E, \sigma)$; $\mu = \{(n, s) \mid n \in V, s \in \{+, -\}\}$
Ensure: *pred*, a set of predictions in case of consistency, or I, an inconsistent graph
  $h \Leftarrow$ BQ.core($\mathcal{N}$,μ)
  if BQ.consistency(h,μ) is **true then**
    pred $\Leftarrow$ BQ.predictions($\mathcal{N}$,μ)
    return *pred*
  else
    $I \Leftarrow$ BQ.inconsistentSubgraph(h,μ)
    return I
  end if

We designed a second program that approximated[2] all the inconsistent graphs in a network in order to predict the variations of network products over a consistent region of it (see Algorithm 2). The objective of this program was to propose an automatic guidance of the diagnostic when many inconsistent subgraphs were found. This program proposes automatic corrections, which depend on the order of the reported inconsistencies, which in their turn depend on the order of the variables in the TDD. In this solution we prioritize the dataset of observations over the signs in the graph. We will see in Section 3.2 other more equitable, though costly in time, solutions. We applied Algorithm 2 to the *E. coli* network mentioned previously wrt a microarray dataset composed of 255 observations, the results were published in [GBMS09]. The computation time was of 877 s ($\approx$ 14 min).

The Bioquali Python package containing these programs is publicly available at `http://genoweb1.irisa.fr/Serveur-GPO/outils/help/datafiles_test/BioPyquali.tgz`.

[1] All CPU times indicated in Section 3.1 were obtained on a Linux PC equipped with Intel Core2 2.16GHZ processor and 2GB of main memory.

[2] See Section 3.2 to find exactly all the inconsistencies.

Algorithm 2 Approximate all inconsistent subgraphs

Require: $\mathcal{N} = (V, E, \sigma)$; $\mu = \{(n, s) \mid n \in V, s \in \{+, -\}\}$
Ensure: *pred*, a set of predictions after removing the inconsistencies, and *incE*, a set of all the inconsistent edges reported
  $h \Leftarrow$ BQ.core($\mathcal{N}$,μ)
  while BQ.consistency(h,μ) is **false do**
    $I \Leftarrow$ BQ.inconsistentSubgraph(h,μ)
    for $(a, b) \in I$.edges() **do**
      $\mathcal{N}$.edgeSign$(a, b) \Leftarrow$ **?**
      $incE \Leftarrow incE \cup \{(a, b)\}$
    end for
    $h \Leftarrow$ BQ.core($\mathcal{N}$,μ)
  end while
  $pred \Leftarrow$ BQ.predictions($\mathcal{N}$,μ)
  return (*pred*,*incE*)

3.1.2.2 Inferring the TF roles in unsigned TRNs

A second problem approached using the Igraph Python module of the Bioquali package was the TF role inference problem. To solve it we built a qualitative system of constraints from the biological data as explained in Section 2.4. There are two main differences wrt the previous analysis that influence computation time: *(i)* the number of variables, and thus the size of the TDD, is given by the number of unknown edges in the graph and *(ii)* there is not a reduction technique to cut the number of edge variables. This means that for inferring the roles of the interactions of the *E. coli* network shown in Section 3.1.2.1, the system of constraints will have around 5000 variables. In our previous analysis our system of constraints had 82 variables, which was the number of non-observed nodes after the reduction technique. For that reason, the analysis had to be simplified to overcome the memory complexity. These simplifications implied that not all of the role predictions could be reported.

In Algorithm 3 we show the steps to automatically infer the roles of TFs in an unsigned network by using the Bioquali functionalities described in Section 3.1.1.2. In order to handle the large number of variables we divided the graph into motifs and checked the *edge consistency* of each motif separately. In this analysis, inconsistent edges are *isolated* (labeled as **?**), and are not used to compute the predictions. This correction step is a way to deal with inconsistencies automatically, in which priority is given to the datasets of observations. After running this program we may stay with a set of predictions of the roles of the network interactions. This set can be afterwards ranked according to the number of datasets that allowed us to infer an interaction role. In this way we can assign a weight to each prediction. Furthermore, we also obtain a list of inconsistencies after confronting the network topology with multiple datasets.

In [VGLB$^+$08] we applied Algorithm 3 to the transcriptional networks of *E. coli* (1415 nodes, 2899 edges) and *S. cerevisiae* (2419 nodes, and 4344 edges); initially, with all their interactions unsigned. The results of this analysis will be shown in Chapters 5 and 6. The time performance was of 801 s ($\approx$ 13 min) and 4335 s ($\approx$ 72 min)

Algorithm 3 Infer the TF roles of an unsigned network

Require: $\mathcal{N} = (V, E, \sigma)$, an unsigned network; M a set of datasets.
Ensure: $predE$, a set of predicted interaction roles; and $incE$ a set of inconsistent edges.

```
incE ⇐ φ
repeat
  numPredE ⇐ 0; N2 ⇐ N
  for all n ∈ V do
    S ⇐ N.subgraph(n,N.predecessors(n))
    if BQ.edgeConsistency(S,M) is true then
      predE ⇐ BQ.edgePredictions(S,M)
      for (a,b,s) ∈ predE do
        N2.edgeSign(a,b) ⇐ s
        numPredE ⇐ numPredE + 1
      end for
    else
      for (a,b) ∈ S.edges() do
        N2.edgeSign(a,b) ⇐ ?
        incE ⇐ incE ∪ {(a,b)}
      end for
    end if
  end for
  for all μ ∈ M do
    (pred,inc) ⇐ Algorithm2(N2,μ)
    for (a,b) ∈ inc do
      N2.edgeSign(a,b) ⇐ ?
    end for
    incE ⇐ incE ∪ inc
    μ ⇐ μ ∪ pred
  end for
  N ⇐ N2
until numPredE is 0
predE ⇐ φ
for (a,b) ∈ N.edges() do
  s ⇐ N.edgeSign(a,b)
  if s ∈ {+,−} then
    predE ⇐ predE ∪ {(a,b,s)}
  end if
end for
return (predE,incE)
```

for the *E. coli* and *S. cerevisiae* networks, respectively. The Bioquali Python package with the TF role inference functionality is publicly available at `http://www.irisa.fr/symbiose/interactionNetworks/data/src/Bioquali_TFrole_Inf.tgz`.

3.1.2.3 Consistency analysis of signed signaling networks

Algorithms 1 and 2, implemented using the Igraph Python module of Bioquali, can analyze hierarchical large-scale transcriptional gene networks, such as *E. coli*, in a rea-

sonable time. This is because the global topology of this network can be reduced to a small system of around 100 variables. Indeed, empirical data indicate that transcriptional networks are sparsely connected, being the number of TF regulators over a gene on average two, while theoretical results show that selection for robust gene networks favors minimally complex and sparsely connected networks [Lec08]. However, using a different level of molecular abstraction, regulatory networks can be represented by more complex systems composed of causal relationships that are not only transcriptional. The signaling network of the human EWS-FLI1 oncogene is a clear example. The computation time of the consistency analysis (Algorithm 1) in this network took more than 7 days using the Igraph Python module. The size of this network, though smaller than *E. coli*, was only reduced to a system of 266 nodes and 596 edges, and had two times more variables than the *E. coli* system. Hence, Algorithms 1 and 2 were reprogrammed using the Dgraph Python module (*cf.* Section 3.1.1.3) in order to analyze signaling networks in a reasonable time.

Taking advantage of the more specific type of representation of regulatory interactions implemented in Bioquali (*cf.* Section 3.1.1.3), we propose two methods that reason over the obtained predictions, as it was shown in the red block of the flowchart in Fig. 2.8.

How to explain a known $\{+,-\}$ variation of a network product? As seen in the first chapter, collecting regulatory network information is a difficult task, which involves sequence treatment, databases analyses, and literature searching among others steps. Once the roles of TFs are known, it may occur that many regulations arrive to a node of interest. It frequently happens that the most important node in the regulatory model will receive more influences, as more articles, experiments, or sequence analyses were considered around this particular node. A natural question in this type of situation is to know by which pathway an observed node was activated or inhibited. We propose a program to answer to this question that provides an explanation to an observed variation of a network product. It performs a post-analysis of the predictions obtained after the consistency analysis, which allows to backtrack in the network until the possible origins of an observed change. In this way we can discover which of the multiple pathways arriving to a node agree or disagree with its observed variation.

In Algorithm 4 we provide the steps implemented in this program. All the Bioquali functionalities are the same as used in Algorithms 1 and 2, implemented using the Dgraph Python module. The new step in this analysis, that takes advantage of the extension of the generic rule, is *addFuncPredec($\mathcal{N}$,n,GEN)*. This function adds a temporary layer between the node n and its direct predecessors, modeling the new influences coming from this layer to n with the generic function GEN_n (recall Equation 2.4). This will be used to easily identify which influences were predicted to change in the same (or opposite) direction as the node of interest.

We tested Algorithm 4 on the signaling network of the EWS-FLI1 human oncogene (296 nodes – 36 receiving a non-generic rule, 430 $\{+,-\}$ edges) with a dataset of 61 observations. The computation time, using the Dgraph Python module of the Bioquali

library, was of 86 s. The obtained results will be presented in Chapter 6.

Algorithm 4 Find the minimal subgraph that explains the $\{+,-\}$ sign of a node in the network

Require: $\mathcal{N} = (V, E, \sigma)$: an oriented graph with V vertexes, E edges, and $\sigma = \{F_T \mid T \in V\}$ as the set of combined influences for each network node
$\mu = \{(n, s) \mid n \in V, s \in \{+,-\}\}$: a partial valuation of the nodes of $\mathcal{N}$
$n \in V$: a node of interest
BQ.consistency($\mathcal{N}$,μ) is **true**
Ensure: $T = (V_T, E_T), V_T \subset V, E_T \subset E$
$T \Leftarrow \phi$
$\mathcal{N}2 \Leftarrow$ addFuncPredec($\mathcal{N}, n$,GEN)
$fixed \Leftarrow$ BQ.predictions($\mathcal{N}2$,μ)
for p in $\mathcal{N}2$.predecessors(n) **do**
  if $p \in dom(fixed)$ **then**
    $\overline{\mu} \Leftarrow \mu$; $\overline{\mu}(p) \Leftarrow \neg fixed(p)$
    $i \Leftarrow$ BQ.inconsistentSubgraph($\mathcal{N}2$,$\overline{\mu}$)
    T.add(i)
  end if
end for

Which nodes to fix to explain a known $\{+,-\}$ variation of a network product? In some cases, after applying Algorithm 4, we may obtain an empty graph; thus, it is not possible to conclude which dataset of observations could explain the observed variation of our node of interest, nor by which pathways this influence arrives to our node of interest. This can be a consequence of a generic model, which does not allow to predict the fluctuation of a node when opposite-sign influences arrive to it.

To overcome this problem we propose a second method that answers to the following question: Which network (non-observed, non-predicted) product has to be fixed to provide a path of influences that explains a known variation of a node of interest without contradicting the known observations? This method is detailed in Algorithm 5. The function $addFuncPredec(\mathcal{N}, n, AND/OR)$ is similar to the one applied in Algorithm 4: it adds a temporary layer between the node n and its direct predecessors, modeling the new influences coming from this layer to n with the functions AND_n or OR_n (recall Equations 2.6 and 2.7).

We tested Algorithm 5 on the same EWS-FLI1 signaling network as previously, but with a different dataset composed of 61 observations. The computation time, using the Dgraph Python module of the Bioquali library, was of 103 s. The obtained results will be presented in Chapter 6.

Algorithm 5 Find a list of pairs $\{(m,s) \mid m \in V, s \in \{+,-\}\}$ that explain a fixed $\{+,-\}$ variation of a node of interest n. Find the subgraph T connecting each node in the list to n

Require: $\mathcal{N} = (V,E,\sigma)$: an oriented graph with V vertexes, E edges, and $\sigma = \{F_T \mid T \in V\}$, the set of combined influences for each network node
$\mu = \{(n,s) \mid n \in V, s \in \{+,-\}\}$: a partial valuation of the nodes of $\mathcal{N}$
$n \in V$: a node of interest
$sign \in \{+,-\}$: the sign of n to be explained
BQ.consistency($\mathcal{N}$,μ) is **true**
Algorithm4(($\mathcal{N}$,μ,n) is ϕ
Ensure: $\{(m,s)\}, T = (V_T, E_T), V_T \subset V, E_T \subset E$
$\{(m,s)\} \Leftarrow ; T \Leftarrow \phi$
$pred1 \Leftarrow$ BQ.predictions($\mathcal{N}$,μ)
if *sign* is '+' **then**
$\mathcal{N}2 \Leftarrow$ addFuncPredec($\mathcal{N}$,n,AND)
else
$\mathcal{N}2 \Leftarrow$ addFuncPredec($\mathcal{N}$,n,OR)
end if
$\mu_2 \Leftarrow \mu$; $\mu_2(n) \Leftarrow sign$
$pred2 \Leftarrow$ BQ.predictions($\mathcal{N}2$,μ_2)
$\{(m,s)\} \Leftarrow pred2 \setminus pred1$
for $(\overline{m}, \overline{s})$ in {(m,s)} **do**
$\mu_3 \Leftarrow \mu_2$; $\mu_3(\overline{m}) \Leftarrow \neg\overline{s}$
$i \Leftarrow$ BQ.inconsistentSubgraph($\mathcal{N}2$,μ_3)
T.add(i)
end for

3.1.3 Computational time synthesis

3.1.3.1 Igraph

In Table 3.2 we present the summary of computation time of the three analyses proposed using Algorithms 1, 2, and 3 implemented using the Igraph Python module of the Bioquali package. In all the cases we deal with large-scale networks, being the consistency check process the fastest. The computation time of Algorithms 2 and 3 will depend on the number of inconsistencies found.

Table 3.2: Computation time for Algorithms 1, 2, and 3 using different biological data. All the analyses were performed on a Linux PC equipped with Intel Core2 2.16GHZ processor and 2GB of main memory. All networks are provided with their number of nodes (n) and edges (e).

Network	Dataset(s)	Analysis	Time (sec.)
E. coli 1915n - 5140e	1 dataset of 45 obs.	Consistency check (Alg. 1) Diagnosis	7
E. coli 1915n - 5140e	1 dataset of 45 obs.	Consistency check (Alg. 1) Predictions	11
E. coli 1915n - 5140e	1 dataset of 255 obs.	Approximate all inconsistencies (Alg. 2)	877
E. coli 1415n - 2899e	61 datasets with 97 obs. in average	Infer TF roles (Alg. 3)	801
S. cerevisiae 2419n - 4344e	15 datasets with 735 obs. in average	Infer TF roles (Alg. 3)	4335

3.1.3.2 Dgraph vs. Igraph

In Table 3.3 we show a comparison between the analyses programmed using Igraph and Dgraph. We see that for the *E. coli* transcriptional network the computation time differs but remains bearable, while for the EWS-FLI1 signaling network the difference in computation time increases considerably.

A web form is publicly available at `http://www.irisa.fr/symbiose/bioquali/`. It uses the implementation of Bioquali based on dependency graphs to compute the consistency check process. We made available this tool in order to encourage prospective users from other disciplines to perform their analyses in a more constrained but easier environment. In addition, Algorithms 1 and 2 (based on Dgraph) are also included in a Cytoscape plugin named BioQualiPlugin and as a Web service. In Chapter 4 we will give more insights on these bioinformatic tools.

3.2 Using ASP to solve qualitative constraints

Reasoning with Answer Set Programming (ASP) appeared to be an interesting alternative to implement the consistency check analysis of large-scale networks. This strategy does not study (nor represent in memory) the set of all solutions, but applies search

Table 3.3: Computation time for Algorithms 1 and 2 programmed using the Igraph and Dgraph Python modules. The analyses were performed on a Linux PC equipped with Intel Core2 2.16GHZ processor and 2GB of main memory. The networks, as well as their cores, are provided with their number of nodes (n) and edges (e).

Network	**Dataset**	**Core**	**Analysis**	**Igraph (s.)**	**Dgraph (s.)**
E. coli 1915n - 5140e	45 obs.	127n - 402e	Alg. 1 Diagnosis	7	3
E. coli 1915n - 5140e	45 obs.	127n - 402e	Alg. 1 Predictions	11	3
E. coli 1915n - 5140e	255 obs.	359n - 984e	Alg. 2	877	140
EWS-FLI1 287n - 644e	61 obs.	266n - 596e	Alg. 1 Diagnosis	$>$ 7 days	26
EWS-FLI1 287n - 644e	61 obs.	266n - 596e	Alg. 1 Prediction	$>$ 7 days	10

to decide whether there exists at least one. In this section we will give details on which problems, concerning the confrontation between large-scale networks and high-throughput data, were approached with ASP and how.

3.2.1 ASP logic

ASP [Bar03] provides a leading declarative language for knowledge representation, reasoning, and declarative problem solving. It results from the combination of a rich modeling language of logic programming with respect to the *answer set* semantics [SNS02], and efficient inference engines based on Boolean constraint solving technology, such as *clasp* [GKNS07]. Some of its advantages wrt other (semi-)declarative languages like Prolog are that it allows disjunctions in the head of rules, and that the order of the rules, body literals, and queries does not matter. The way Prolog uses negation as failure to refute a query may cause it to get into infinite loops; ASP, in contrast, does not have this risk as it does not deduce proofs. This difference causes ASP not to get into infinite loops, however, it is not suitable for generating a proof of a property of a model as BIOCHAM [CFS06] does. Nevertheless, different types of complex calculations can be performed, and it suits very well the consistency analysis proposed in Chapter 2.

The idea of ASP is to encode a problem as a set of rules and then find its solution by obtaining the *answer sets* of the program. In the following we will elaborate more this idea. An ASP program is a collection of rules of the form:

$$L_0 \; or \ldots \; or \; L_k \leftarrow L_{k+1}, \ldots, L_m, \mathbf{not} \; L_{m+1}, \ldots, \mathbf{not} \; L_n. \tag{3.2}$$

where each L_i is an atom, '*or*' stands for the logical $\vee$ symbol, and ',' stands for $\wedge$. The strict meaning of '*or*', however, differs from that of $\vee$. This rule means that if $L_{k+1}, \ldots, L_m$ are true and if $L_{m+1}, \ldots, L_n$ can be assumed false, then at least one of

$L_0, \ldots, L_k$ must be true. The parts on the left (resp. right) of '$\leftarrow$' are called *head* (resp. *body*) of the rule. A rule with an empty body and a single disjunct in the head ($k = 0$) is called a *fact*, while a rule with an empty head is called an *integrity constraint*.

An atom is of the form $p(t_1, \ldots, t_n)$, where p is the predicate and each t_i is either a variable, a constant, or an n-ary function. We will say that an atom is *ground*, if there is no variable in it. An interpretation (S) of a program P is any set of ground atoms. S is said to *satisfy* the ASP rule (3.2) if:

(i) when the rule head is empty: $\{L_{k+1}, \ldots, L_m\} \subsetneq S$ or $\{L_{m+1}, \ldots, L_n\} \cap S \neq \phi$, or

(ii) when the rule head is not empty: $\{L_{k+1}, \ldots, L_m\} \subseteq S$ and $\{L_{m+1}, \ldots, L_n\} \cap S = \phi$ implies that $\{L_0, \ldots, L_k\} \cap S \neq \phi$.

A *model* of P is an interpretation that satisfies all the rules in P. *Answer sets* of a program P are particular models of P satisfying an additional criterion, which will be explained in the following lines. Given a set S of ground atoms of P, let P^S be the *reduct* program of P obtained by deleting:

(i) each rule that has an atom **not** L in its body with $L \in S$, and

(ii) each atom **not** $\overline{L}$ in the bodies of the remaining rules.

S will be an *answer set* of program P, iff S is minimal among all the models of P^S. Answer sets are defined on ground programs; however, ASP modeling language allows for non-ground problems encodings. Grounders, like *gringo* [GST07] and *lparse* [Syr], are capable of combining a problem encoding and ground facts into an equivalent ground program, processed by some ASP solver.

In [GST+08] Gebser and colleagues approached the consistency analysis using ASP. They proposed ASP rules that could decide if a network was consistent or not wrt a dataset, as well as rules to compute all inconsistent subgraphs. Another question that was approached with ASP is how to minimally correct an inconsistent network wrt a dataset. In the next sections we will show part of the ASP encodings proposed for these analyses.

3.2.2 Consistency check and diagnosis

The consistency check of a network $\mathcal{N}$ wrt to a dataset μ was analyzed in [GST+08] in terms of finding the answer set that satisfies a set of rules of type 3.2. If no answer set was found, then $\mathcal{N}$ was inconsistent wrt μ. Let us recall our previous network and dataset notations:

- Let us denote $\mathcal{N} = (V, E, \sigma)$, the oriented and signed network to be analyzed, where V is the set of nodes, E the set of edges, and $\sigma : E \rightarrow \{+, -\}$, a labeling of the edges in the network.

- Let $\mu = \{(n, s) \mid n \in V, s \in \{+, -\}\}$ represent the dataset of partial significant variations of the nodes in $\mathcal{N}$.

The ASP rules proposed in [GST+08] to determine the consistency of a network $\mathcal{N}$ wrt a single dataset μ were divided in three parts:

Problem Instance. These rules were ground facts built from the information contained in $\mathcal{N}$ and μ, that is:

- Network facts: for each $i \in V$ a fact of the form `'vertex(i)'` is added, for each edge $(i, j) \in E$ the fact `'edge(i,j)'` is added, and for each labeled edge $\sigma(i, j) = s$ the fact `'obsE(i,j,s)'` is added. If i is a node without predecessors in the network, then the fact `'input(i)'` is added.
- Dataset facts: for each $(i, s) \in \mu$ the fact `'obsV(i,s)'` is added.

Generating solution candidates. Let us briefly go back to solving the qualitative system of constraints representing a network $\mathcal{N}$ and a dataset μ. In Section 2.2 we explained that the qualitative system will be consistent if there exists at least one $\{+, -\}$ valuation of all the nodes in the network such that all constraints in the system are satisfied. Thus, in order to derive all the solution candidates with ASP, it was necessary to derive all the possible $\{+, -\}$ valuations of all the network nodes (and edges). This was done by adding the following rules to the program.

$$\begin{aligned} labelV(V, +) \text{ or } labelV(V, -) &\leftarrow vertex(V). \\ labelE(U, V, +) \text{ or } labelE(U, V, -) &\leftarrow edge(U, V). \\ labelV(V, S) &\leftarrow obsV(V, S). \\ labelE(U, V, S) &\leftarrow obsE(U, V, S). \end{aligned} \tag{3.3}$$

Note that if the dataset is composed of the observation $\mu(i) = +$, the minimal criteria of the answer set computed following the rules in (3.3) will only include the atom `'labelV(i,+)'`. This happens because an instantiation including both atoms `'{labelV(i,+),labelV(i,-)}'` will be redundant. This is a way of fixing the observation values; the same applies to edge labels.

Testing solution candidates. The consistency rule proposed in Chapter 2 stated that: *The variation of the concentration level of one molecule in the network must be explained by an influence received from at least one of its predecessors, different from itself, in the network.* In [GST+08] the authors proposed an analogous definition of consistency, which was: given $\mathcal{N}$ and μ, *for every non-input vertex $i \in V$, the sign $\mu(i)$ of i is consistent if there is some edge $(j, i) \in E$ such that $\mu(i) = \mu(j) \otimes \sigma(j, i)$.*

Following this definition, in order to decide if a set of facts representing the network and dataset was consistent, three more rules were added. The first two derived the atom `'receive(i,s)'` if node i received an influence of sign s:

$$\begin{aligned} receive(V, +) &\leftarrow labelE(U, V, S), labelV(U, S). \\ receive(V, -) &\leftarrow labelE(U, V, S), labelV(U, T), S \neq T. \end{aligned} \tag{3.4}$$

The last rule eliminated all solution candidates that were not inputs and did not receive an influence of the same sign.

$$\leftarrow labelV(V,S), \textbf{not } receive(V,S), \textbf{not } input(V). \tag{3.5}$$

If there is at least one answer set of this program, then $\mathcal{N}$ is consistent wrt μ.

Another set of rules was proposed in order to perform the diagnosis step and to find exactly all the minimal inconsistent subgraphs. The encodings and examples of these analyses are given in [GST+08]. Also, they provided a web site [asp] from which both analyses can be launched on-line on their machines. In [GST+08] the authors applied this analysis to the TRN of *S. cerevisiae* [NGBBK02] (491 nodes and 909 edges) wrt a dataset of 20 observations, obtaining all the inconsistent subgraphs in 218 s. We also applied this approach to the transcriptional *E. coli* network (1915 nodes and 5140 edges) wrt a dataset of 255 observations. The computation time of finding exactly all the inconsistent subgraphs was longer, however, this analysis provided interesting results and they will be presented in Chapter 5.

3.2.3 Minimal network/dataset repairs

We address now a new problem, which is to repair large-scale biological networks and corresponding, yet often discrepant, measurements in order to predict unobserved variations. To this end, we proposed a range of different operations for altering the experimental dataset and/or the biological network in order to re-establish their mutual consistency. We submitted part of this work, in collaboration with the team leaded by T. Schaub in the Potsdam University, to the "Twelfth International Conference on the Principles of Knowledge Representation and Reasoning", to be held in Toronto, Canada, May 9-13, 2010.

This framework was validated by an empirical study on the *E. coli* TRN (1915 nodes and 5140 edges) wrt two datasets: Exponential-Stationary growth shift [BMA+07] and Heatshock [AHL+03], composed of 850 significant observations. We randomly selected samples of size corresponding to 3%, 6%, 9%, 12%, and 15% of the whole data, and used them for testing both our repair modes as well as prediction (of omitted data). The ASP encodings proposed to solve this problem, as well as the computation time to perform the repair and prediction analyses, will be presented in the following subsections. The validation of the predictions obtained will be presented afterwards in Chapter 5.

3.2.3.1 Problem instance

The network and dataset are encoded as ground facts with the same rules used for the consistency check analysis (*cf.* Section 3.2.2). However, to ease the rules formulation, the signs in network $\mathcal{N}$ and dataset μ are given by $\{1, -1\}$ values instead of $\{+, -\}$. Let us show an example of how a network and dataset are coded into ASP rules.

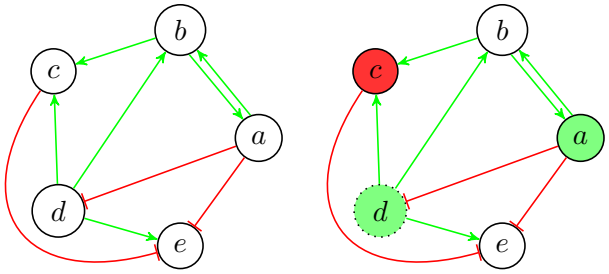


Figure 3.2: An influence graph (left) along with one experimental dataset (right), in which increases (decreases) were observed for vertexes colored green (red), and vertex d is an input. Green (red) edges in the graph represent activations (inhibitions).

Example 1. *The facts describing the influence graph (Π_g) and the experimental dataset (Π_{p_1}), shown in Figure 3.2, are as follows:*

$$\Pi_g = \left\{\begin{array}{llll} vertex(a). & vertex(b). & vertex(c). & vertex(d). \quad vertex(e). \\ edge(a,b). & obsE(a,b,1). & edge(c,e). & obsE(c,e,-1). \\ edge(a,d). & obsE(a,d,-1). & edge(d,b). & obsE(d,b,1). \\ edge(a,e). & obsE(a,e,-1). & edge(d,c). & obsE(d,c,1). \\ edge(b,a). & obsE(b,a,1). & edge(d,e). & obsE(d,e,1). \\ edge(b,c). & obsE(b,c,1). & & \end{array}\right\} \quad (3.6)$$

$$\Pi_{p_1} = \{input(d).\, obsV(d,1).\, obsV(c,-1).\, obsV(a,1).\} \quad (3.7)$$

*Note that experimental profile p_1 (*cf. *right in Figure 3.2) is inconsistent with the given influence graph. It necessitates labeling vertex b with -1 in order to explain the observed decrease of c. In p_1 such a decrease of b is unexplained.* ◇

3.2.3.2 Repair operations

These rules state the conditions necessary to be fulfilled in order to apply a specific repair. Before introducing them, we describe in Table 3.4 the provided possible repairs.

Table 3.4: Network and dataset repairs proposed in order to reestablish their mutual consistency.

Term	Target	Meaning
add_e(U,V)	network	Introduce an edge from U to V
flip_e(U,V,S)	network	Flip the sign S of the existing edge from U to V
inp_v(V)	network	Treat vertex V as an input
flip_v(V,S)	dataset	Flip the sign S of vertex V

These repair operations are inspired by existing biological use cases. To repair a model by adding new edges makes sense when the model is incomplete (which is often

the case in practice). Flipping the sign of an edge is a way to curate the model; it means that in some experiment the effect of a regulator (activator or inhibitor) should be corrected. Turning a vertex into an input can be used to indicate missing (unknown) regulations or oscillations of regulators. Revising experimental observations puts the dataset into question and may help to identify aberrant measurements (frequent in microarray data).

The following rules were used to define admissible repair operations:

$$\begin{array}{ll} rep(add\_e(U,V)) & \leftarrow rep\_a, vertex(U), vertex(V), U \neq V, \textbf{not } edge(U,V). \\ rep(flip\_e(U,V,S)) & \leftarrow rep\_e, edge(U,V), obsE(U,V,S). \\ rep(inp\_v(V)) & \leftarrow rep\_i, vertex(V), \textbf{not } input(V). \\ rep(flip\_v(V,S)) & \leftarrow rep\_v, vertex(V), obsV(V,S). \end{array} \tag{3.8}$$

Observe that particular operations are identified with *function terms* inside of predicate *rep*, which enables us to deal with repairs in a general way whenever knowing particular types is unnecessary.

Which repair operations ought to be permitted or omitted requires background knowledge about the model and data at hand. By offering a variety of operations, our framework is flexible and may be adjusted to particular situations. In (3.8) the declaration of admissible repair operations is governed by atoms $rep\_a, \ldots, rep\_v$. Depending on the requested repair types, such atoms are to be provided as facts. It would also be possible to restrict repair operations to particular edges or vertexes, respectively, based on the availability of biological expert knowledge.

Finally, note that the rules in (3.8) filter some redundant repairs. An edge between distinct vertexes can be introduced only if there is none in the model. Flipping signs of edges or vertexes is possible only if a sign is provided in the model or the data, respectively. Making a vertex an input requires it not to already be defined as an input.

3.2.3.3 Repair encoding

With admissible repairs at hand, the following rules encode the choice of operations to apply:

$$app(R) \leftarrow rep(R), \textbf{not } \overline{app}(R). \qquad \overline{app}(R) \leftarrow rep(R), \textbf{not } app(R). \tag{3.9}$$

The rest of the repair encoding is about identifying witnesses for the consistency of the repaired network/dataset. We first declare available signs and their relationships:

$$sig(1). \quad sig(-1). \qquad\qquad opp(S,-S) \leftarrow sig(S). \tag{3.10}$$

The rules presented below take care of labeling edges and also incorporate repairs on them:

$$\begin{array}{l} labelE(U,V,S) \leftarrow edge(U,V), obsE(U,V,S), \textbf{not } app(flip\_e(U,V,S)). \\ labelE(U,V,T) \leftarrow app(flip\_e(U,V,S)), opp(S,T). \\ labelE(U,V,S) \leftarrow app(add\_e(U,V)), opp(S,T), \textbf{not } labelE(U,V,T). \\ labelE(U,V,S) \leftarrow edge(U,V), opp(S,T), \textbf{not } labelE(U,V,T). \end{array} \tag{3.11}$$

The first rule is to preserve (known) signs of edges if not flipped by a repair; otherwise, the second rule is used to derive the opposite sign instead. For edges introduced by repairs and unlabeled edges in the model, respectively, the last two rules encode the choice of a sign, making sure that any answer set comprises a total edge labeling, given by ground atoms over predicate *labelE*.

Using the same methodology as with edges, but now relative to experimental profiles, the following rules deal with vertex labels and repairs on them:

$$\begin{aligned} &\mathit{label}\,V(V,S) \leftarrow \mathit{vertex}(V), \mathit{obs}\,V(V,S), \mathbf{not}\ \mathit{app}(\mathit{flip\_v}(V,S)).\\ &\mathit{label}\,V(V,T) \leftarrow \mathit{app}(\mathit{flip\_v}(V,S)), \mathit{opp}(S,T).\\ &\mathit{label}\,V(V,S) \leftarrow \mathit{vertex}(V), \mathit{opp}(S,T), \mathbf{not}\ \mathit{label}\,V(V,T). \end{aligned} \tag{3.12}$$

Again, the first rule maintains signs given in an experimental dataset, while the second rule applies repairs flipping such signs. Ground instances of the third rule permit choosing signs of unobserved vertexes, not yet handled by either the first or the second rule. As a consequence, the instances of *labelV* in an answer set provide a total vertex labeling.

Finally, we need to check whether the variations of all non-input vertexes are explained by the influences of their regulators. This is accomplished as follows:

$$\begin{aligned} \mathit{receive}(V, S{*}T) &\leftarrow \mathit{labelE}(U,V,S), \mathit{label}\,V(U,T).\\ &\leftarrow \mathit{label}\,V(V,S), \mathbf{not}\ \mathit{receive}(V,S),\\ &\quad\ \mathbf{not}\ \mathit{input}(V), \mathbf{not}\ \mathit{app}(\mathit{inp\_v}(V)). \end{aligned} \tag{3.13}$$

First, observe that the influence of a regulator U on V is simply the product of the signs of the edge and of U. Based on this, the integrity constraint denies cases where a non-input vertex, neither a given input nor made input by any repair, receives no influence compatible with its variation S. That is, a non-input vertex must not be unexplained in the dataset. Conversely, any answer set comprises consistent total vertex and edge labellings wrt the repaired network/dataset.

3.2.3.4 Minimal repairs

Typically, plenty of repairs are possible, in particular, if several repair operations are admitted by adding multiple control atoms $\mathit{rep\_a}, \ldots, \mathit{rep\_v}$ as facts. However, one usually is only interested in repairs that make few changes on the model and/or data.

Repairs that achieve consistency by applying a minimal number of operations can easily be selected among the candidate repairs using the $\#\mathit{minimize}$ directive available in *lparse*'s and *gringo*'s input languages. The respective statement is as follows:

$$\#\mathit{minimize}\{\mathit{app}(R) : \mathit{rep}(R)\}. \tag{3.14}$$

It means that the number of instances of predicate *app* in answer sets, with argument R ranging over the ground instances of "domain predicate" *rep*, is subject to minimization. Note that (3.14) does not explicitly refer to the types of repair operations to be minimized.

Using the *E. coli* network and the two datasets, we tested the feasibility of our repair modes on consistent as well as inconsistent samples (depending on the random selection). Table 3.5 provides average run-times and numbers of timeouts in parentheses over 200 samples per percentage of preassigned measurements; timeouts are included as 600 s in average run-times. We ran experiments admitting the following repair operations and combinations thereof: flipping edge labels denoted by 'e' (*flip_e*), making vertexes input denoted by 'i' (*inp_v*), and flipping preassigned variations denoted by 'v' (*flip_v*).

We do not include results on the adding edges repair (*add_e*), where the bottleneck lies in grounding since permitting the addition of arbitrary edges turns the influence graph into a large clique at the encoding level. To avoid this, the edges that can possibly be added by repairs should be restricted to a (smaller) set of reasonable ones, which requires biological knowledge, *e.g.*, regulations known for organisms related to the one under consideration.

Table 3.5: Computation times for minimal repairs tested on the *E. coli* TRN (1915 nodes and 5140 edges) wrt two datasets composed of 850 significant observations. 200 samples of size corresponding to 3%, 6%, 9%, 12%, and 15% of the whole data were randomly selected for the experiments. The analysis was run with grounder *gringo* (2.0.3) and solver *clasp* (1.2.1) on a Linux PC equipped with AthMP+1900 processor and 4GB of main memory, imposing a maximum time of 600 seconds per run. The numbers in parentheses represent the number of timeouts.

	Exponential vs Stationary					Heatshock				
Repair	3%	6%	9%	12%	15%	3%	6%	9%	12%	15%
e	6.58(0)	8.44 (0)	11.60 (0)	14.88 (0)	26.20 (0)	25.54 (4)	42.76 (8)	50.46 (5)	69.23 (6)	84.77 (6)
i	2.18(0)	2.15 (0)	2.21 (0)	2.23 (0)	2.21 (0)	2.10 (0)	2.13 (0)	2.13 (0)	2.05 (0)	2.08 (0)
v	1.41(0)	1.40 (0)	1.40 (0)	1.41 (0)	1.37 (0)	1.41 (0)	1.47 (0)	1.42 (0)	1.37 (0)	1.39 (0)
e i	73.16(6)	202.66(23)	392.97 (87)	518.50(143)	574.85(179)	120.91(21)	374.69 (91)	553.00(169)	593.20(197)	595.99(198)
e v	28.53(0)	85.17 (0)	189.27 (12)	327.98 (33)	470.48 (88)	67.92 (3)	236.05 (31)	465.92(107)	579.88(179)	596.17(197)
i v	2.09(0)	2.14 (0)	2.45 (0)	3.08 (0)	6.06 (0)	2.27 (0)	4.94 (0)	60.63 (8)	257.68 (56)	418.93(123)
e i v	133.84(8)	391.60(76)	538.93(151)	593.33(193)	600.00(200)	232.29(26)	542.48(152)	593.88(195)	600.00(200)	600.00(200)

3.2.3.5 Prediction under repairs

The last step of our analysis was to provide $\{1, -1\}$ predictions for some network nodes after the network and dataset were corrected performing minimal repairs. In the ASP framework, enumerating all consistent total labellings in order to do prediction is unnecessary. In fact, given a program Π, the intersection of all (optimal) answer sets of Π can be computed using Algorithm 6[3].

When a node of the network i is predicted to a value -1, all the answer sets of Π will contain the labeling '`labelV(i,-1)`'. However, when a node is not predicted, only some answer sets will contain the labeling '`labelV(i,-1)`', while others will contain '`labelV(i,1)`'. Thus, non of these labellings will belong to the intersection of all answer

[3]The Herbrand Base of a program Π is the set of all its ground atoms.

Algorithm 6 Computing the intersection of all optimal answer sets of a program Π

Require: Π, a set of ASP rules
Ensure: C, intersection of all optimal answer sets of Π
  $C \Leftarrow$ Herbrand Base of Π
  while there is an answer set X of $\Pi \cup \{\leftarrow C.\}$ **do**
    $C \Leftarrow C \cap X$
  end while
  return C

sets. Observe that, due to intersecting C with an answer set X in Algorithm 6, the contents of C is monotonically decreasing over iterations. Furthermore, augmenting Π with integrity constraint $\leftarrow C.$ makes sure that any remaining answer set X exclude some atom common to all previously computed answer sets.

For prediction, an input program Π is composed of an instance (*cf.* Section 3.2.3.1), a definition of admissible repair operations (*cf.* Section 3.2.3.2), the basic repair encoding (*cf.* Section 3.2.3.3), and the #*minimize* statement in Section 3.2.3.4. Predicted signs for vertexes are then simply read off from instances of predicates *labelV* in the intersection C (Alg. 6) of all optimal answer sets of Π.

The computation time for the prediction under repair is given in Table 3.6. We observe that the run-times for prediction are in line with the ones for computing a cardinality-minimal repair, and maximum time is only rarely exceeded on the samples with known optimum. This shows that prediction is successfully applicable if computing a cardinality-minimal repair is feasible.

Table 3.6: Computation times for prediction under repairs tested on the *E. coli* TRN (1915 nodes and 5140 edges) wrt two datasets composed of 850 significant observations. 200 samples of size corresponding to 3%, 6%, 9%, 12%, and 15% of the whole data were randomly selected for the experiments. The analysis was run with grounder *gringo* (2.0.3) and solver *clasp* (1.2.1) on a Linux PC equipped with AthMP+1900 processor and 4GB of main memory, imposing a maximum time of 600 seconds per run. The numbers in parentheses represent the number of timeouts.

	Exponential vs Stationary					Heatshock				
Repair	3%	6%	9%	12%	15%	3%	6%	9%	12%	15%
e	13.27(0)	12.19(0)	14.76(0)	15.34 (0)	25.90 (1)	25.77(0)	37.18(0)	29.09(0)	36.23(0)	41.88(0)
i	6.18(0)	5.26(0)	4.77(0)	4.60 (0)	4.42 (0)	6.57(0)	5.93(0)	5.17(0)	4.86(0)	4.54(0)
v	4.64(0)	4.45(0)	4.39(0)	4.40 (0)	4.30 (0)	4.86(0)	5.06(0)	5.34(0)	5.42(0)	5.52(0)
e i	35.25(0)	97.66(1)	293.80(3)	456.55 (3)	550.33 (1)	85.47(0)	293.28(1)	524.19(3)	591.81(0)	594.74(0)
e v	14.35(0)	26.17(0)	90.17(3)	200.25(13)	363.36(16)	23.32(0)	111.99(0)	338.95(0)	545.56(2)	591.23(0)
i v	6.43(0)	5.75(0)	6.27(0)	6.69 (0)	8.61 (0)	6.91(0)	6.63(0)	30.33(0)	176.14(1)	371.95(0)
e i v	42.51(0)	248.30(1)	468.71(2)	579.58 (0)	— (0)	101.82(1)	466.91(0)	585.64(0)	—(0)	—(0)

3.2.3.6 Discussion

To our knowledge, this work provides the first approach to automatically and globally reason over whole biological networks in order to identify minimal repairs on the network and/or datasets. This approach is fully declarative so that its intended results are independent of computations. This assigns a clear semantics to (minimal) repairs and prediction, and the explicit representation of repair operations allows for reasoning over corrections on either of or both of data and model. The latter distinguishes our approach to repair from the one applied in the area of databases [ABC99], which is limited to data repair. Importantly, available biological knowledge can be incorporated in our framework to improve both the validity of results and computational efficiency.

3.3 Comparing TDDs with ASP

The TDDs and ASP approaches are complementary. Their main difference is how they approach the problem of consistency. The TDDs provide an efficient representation in memory of all the space of solutions of a qualitative system of constraints. By efficiently covering this structure, the predictions and inconsistencies of the system can be obtained. Using the TDDs approach, methods can be proposed to quantify the number of consistent solutions; and thus to generate *relative predictions* over a node. For example, computing the percentage of solutions in which a network node was predicted to a + value. Contrary to this approach, ASP does not represent all the space of solutions in memory, but it generates a model and then asks existential questions over it. By asking particular questions, it could be possible to generate all the space of solutions with ASP. However, this analysis may take longer computation time. Nevertheless, ASP programs are capable of dealing problems of larger size (without considering a reduction step). This idea enlarges the number of questions that can be asked, in particular in the experimental plan design.

Since ASP is a declarative language, programs can be represented (and solved) in an easier way. For example coding a new variation value for a node, such as the null-variation, can be done by adding new rules into the program. In the TDDs approach, this extension required to modify several parts of its implementation, which is not a trivial task for a new user. There are some problems, like the computation of all inconsistent subgraphs, that can only be approached with ASP. On the other hand, current analyses concerning the consistency of influence graphs, where influences can be also represented using Boolean functions, were approached only by programs based on TDDs.

Both, TDDs and ASP, offer a programing language in which the user can define her specific analyses. In this chapter we presented publicly available programs that use both approaches in order to answer our specific problematics concerning the consistency check between large-scale regulatory networks and wide experimental datasets. However, the programs designed under TDDs and ASP can be extended endlessly, as long as the biological questions concerning large-scale data persist. The choice of using one of these approaches will depend on the type of questions and problematics that we will

be confronted with. In Table 3.7 we show a summary of the different functionalities provided by existing programs based on TDDs and/or ASP.

Table 3.7: Functionalities provided by existing programs based on TDDs and/or ASP.

Functionality	**TDDs**	**ASP**	**Comment**
Consistency check	•	•	-
Prediction under consistency	•	•	-
Finding a minimal inconsistent subgraph	•	•	-
Approximating all inconsistent subgraphs	•	•	-
Finding exactly all inconsistent subgraphs		•	Costly in computation time for networks with more than 1000 nodes.
Inferring TF roles in unsigned networks	•	•	Although implemented by both approaches, only ASP can predict exactly all the TF roles. However, the current ASP implementation requires a pre-step computed using TDDs.
Coding influences as Boolean functions	•		-
Reasoning over the generated predictions	•		-
Prediction under minimal repairs		•	-
Web service	•	•	They provide consistency check, a diagnosis, and approximating all inconsistent subgraphs. The ASP Web service also provides finding exactly all inconsistent subgraphs.
Analysis visualization	•		It provides consistency check, a diagnosis, and approximating all inconsistent subgraphs.

Chapter 4

Bioinformatic software

Many efforts were done in order to provide the computationally powerful methodologies, detailed in Chapter 3, as publicly available and user-friendly bioinformatic tools. In this chapter we will present three bioinformatic tools that were developed during this thesis using programs based on the TDD approach. These tools are an on-line Website to perform consistency analyses, a Cytoscape plugin to visualize analyses, and a Web service to integrate our analysis into other bioinformatic tools. All three of them use the GenOuest [gen] high performance computing facility. The biggest effort in the informatic design of these tools was the setting up of the Cytoscape plugin, functionalities of which we will describe in this chapter; we published the complete work in [GBMS09].

4.1 Website

The first bioinformatic tool proposed was an on-line Website available at `www.irisa.fr/symbiose/bioquali/`. This Website accepts as input a network and a dataset coded as text files. The network file is composed of lines in the format '`A -> B [+|-|?]`', with one line per network regulation, which expresses that a network product A regulates B as an activator, repressor, or in a complex manner, respectively. The dataset file is composed of lines in the format '`A = [+|-]`', with one line per observation, which expresses that a network product A was observed to be up- or down-regulated, respectively. The types of results displayed are:

- Consistency check of the network with or without an experimental dataset.
- First diagnose, when inconsistent.
- Prediction of the variation of some network products, when consistent.

These results are obtained by running the program presented in Algorithm 1, implemented using the Dgraph Python module. The on-line web form is located on the GenOuest Web server [gen]. The programs checking the consistency are run on the GenOuest high performance computing facility. In Fig. 4.1 we illustrate an example of its usage.

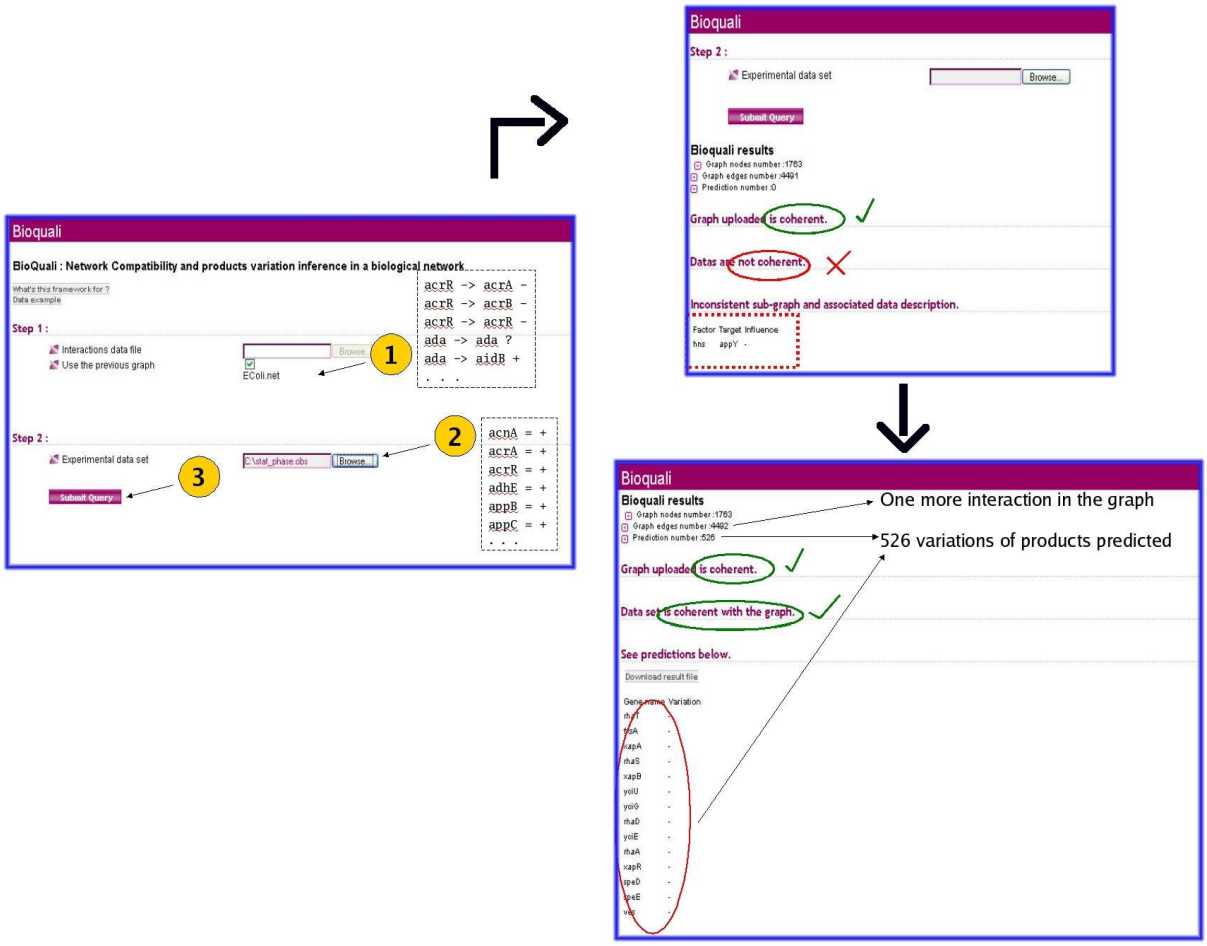


Figure 4.1: A usage example of the Bioquali on-line web form. The left screenshot shows the data initialization. Once the computation is finished the consistency results will appear as shown in the upper-right image. In this example the network itself was consistent, while it was inconsistent wrt the dataset provided. The inconsistent region is shown as a line representing a network regulation. Once the network regulations are corrected, we can launch once more this analysis obtaining this time the results displayed in the bottom-right image, in which both data are consistent and a set of variations of some network products (predictions) is listed.

4.2 Cytoscape plugin

In order to provide a wider range of user interactions with the consistency analysis, we developed a plugin for the Cytoscape environment [SMO+03], designed to facilitate automatic reasoning on regulatory networks. The BioQuali plugin enhances user-friendly conversions of regulatory networks (including reference databases) into signed directed graphs. BioQuali performs automatic global reasoning in order to decide which products in the network need to be up or down regulated (active or inactive) to *globally* explain experimental data. It highlights incomplete regions in the network, meaning that gene

expression levels do not globally correlate with existing knowledge on regulation carried by the topology of the network. The BioQuali plugin facilitates *in silico* exploration of large-scale regulatory networks by combining the user-friendly tools of the Cytoscape environment with high-performance automatic reasoning algorithms. As a main feature, the plugin guides further investigation regarding a system by highlighting regions in the network that are not accurately described and merit specific study.

4.2.1 Implementation

The BioQuali plugin is implemented in Java, based on the Cytoscape API, and uses the REST architectural style. By default, the client component uses an unauthenticated HTTP connection to communicate with the GenOuest Web server [gen]. This enables fast remote execution of the algorithm underlying the BioQuali plugin on the GenOuest high performance computing facility. The GenOuest computing infrastructure consists of 32 AMD Opteron bi-processor nodes (Sun V20Z) with 4GB of main memory each and a job submission server, SGE, which manages access and computation.

The plugin is available to download from the Cytoscape plugin website, under the Plugin/Analysis section. It is packaged as a jar file which must be placed in the Cytoscape plugins directory. It is compiled with the latest Cytoscape API (version 2.6). It can also be installed using the Cytoscape plugin management system, selecting "BioQualiPlugin v.1.1" from the Analysis section. Alternatively, the BioQuali plugin is available via Java Web Start (see [bio]). Documentation and tutorial examples of this plugin are provided at `http://www.irisa.fr/symbiose/projects/bioqualiCytoscapePlugin/`.

4.2.2 BioQuali plugin functionalities

Input

The BioQuali plugin receives two types of input: an experimental dataset and a regulatory network. BioQuali is able to handle large-scale networks, for example, the bacterial transcriptional regulatory networks of *E. coli* [SGCPG*al.*06] and *Corynebacterium* [NBA$^+$07]. However, any type of regulatory network is accepted provided that an annotation file with labeled interactions is first imported to Cytoscape. The plugin provides a user-friendly annotation interface for classifying interaction labels as $\{+, -, \&, ?\}$ regulation types. This classification will then be used to perform automatic reasoning on behaviors. The $\{+, -\}$ types represent positive and negative influences among network products, the '&' is a Boolean *AND* among signs (the variation of a product is positive only when all the influences it receives are positive), and the '?' represents interactions with unclassifiable effects (either unknown or context-dependent sign).

Thanks to the annotation functionality, the plugin is compatible with other network import plugins: the user may use regulatory networks obtained using the CoryneRegNet Cytoscape plugin [BA08] or automatically import the latest update of the RegulonDB database [SGCPG*al.*06] in order to retrieve the *E. coli* transcriptional network.

The experimental dataset, resulting from the comparison of gene or protein expression levels between two conditions, can be provided as raw numbers representing the relative gene expression levels. A functionality of the plugin enables the user to classify these observations as up- or down-regulated $\{+, -\}$ using a chosen threshold. The dataset can also be imported as a Cytoscape node attributes file (.NA) in which certain network products are annotated as $+$ or $-$, depending on their expression change.

Output

The plugin outputs one of the three types of the results after checking for consistency: a list of *local inconsistencies* (LI), a list of *global inconsistencies* (GI), or a *list of predictions*. The first result (LI) is outputted when the network presents inconsistencies of the form: "A is the only activator of B, A increases but B decreases". The second type of inconsistency (GI), much more difficult to detect, is a global one: it is shown as a subgraph in which the sign of its nodes or edges contradicts the flow of events at certain steps (see the example in the next section and illustrated in Fig. 4.2B). The plugin automatically retrieves all the subnetworks of a model that are inconsistent with a certain dataset (iteratively, all the interactions of a GI are fixed to '?' and a next GI is computed). For the third type of result, the plugin outputs a list of predictions when a network is consistent with experimental data: it shows fluctuations in the network products inferred as increasing or decreasing in order to consistently explain the experimental data.

4.2.3 The consistency criteria

The central functionality of the BioQuali plugin is to automatically and visually illustrate the user how to explain her experimental observations regarding the regulatory model. This task is accomplished by using the consistency-check reasoning introduced in Chapter 2 and solved using the TDDs representation described in Chapter 3. In Fig. 4.2 we recall the automatic reasoning underlying the BioQuali plugin. Given a known (signed and oriented) regulatory network in which some products are observed, the plugin reasons over the whole network in order to determine its consistency. In Fig. 4.2A we describe a small regulatory network. Let us say, for example, that *rpsP* and *rpmC* are observed as down-regulated; we then deduce that, as fnr is the only inhibitor of *rpsP*, fnr should be up-regulated. If this is the case, following a similar reasoning, we conclude that *arcA* should be down-regulated. To conclude our analysis, we observe *rpmC* as down-regulated, however, its inhibitor is down-regulated and its activator is up-regulated; we should conclude that *rpmC* is up-regulated, yet its observed change tells us opposite; therefore, we find an inconsistency between model and data (Fig. 4.2B). Using the same network but changing the observed data, *i.e.* *rpsP* up-regulated and *rpmC* down-regulated, leads us through another deduction path, in the case of which it is possible to assign a *unique* $\{+, -\}$ change value to fnr and *arcA* that explains the observed data consistently. This unique deduction is called *prediction* (Fig. 4.2C).

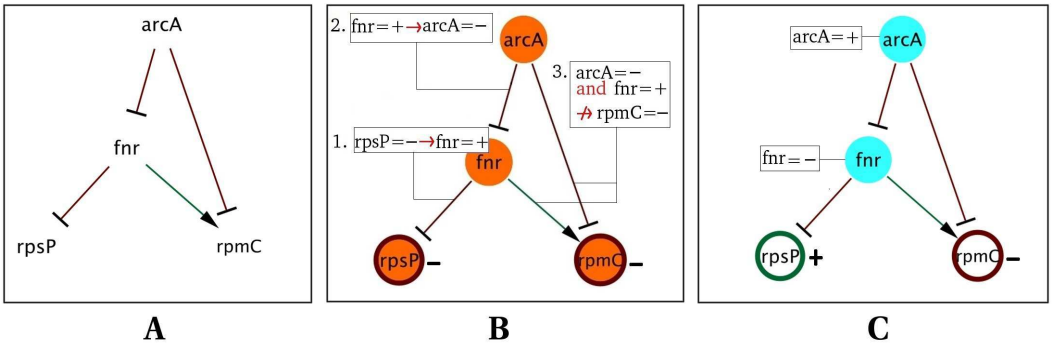


Figure 4.2: Visualizing the consistency criteria. **A.** Signed and oriented regulatory network. Arrows ending with '−>' or '−|' represent activations or inhibitions, respectively. **B.** Detection of a global inconsistency when *rpsP* and *rpmC* are both observed to be down-regulated. **Step 1,** *rpsP*'s negative change implies that its only inhibitor has to be up-regulated ($fnr = +$); **Step 2,** *fnr*'s positive change implies that its only inhibitor has to be down-regulated ($arcA = -$); **Step 3,** these deductions cannot explain *rpmC*'s down-regulation, since its activator (*fnr*) is up-regulated and its inhibitor (*arcA*) is down-regulated. **C.** Prediction when *rpsP* is observed to be up-regulated and *rpmC*, down-regulated. *rpsP*'s inhibitor (*fnr*) is fixed to '−' to explain the observed value of *rpsP*. In a similar manner, *arcA* is fixed to '+'. With this unique configuration we obtain a consistent system where $fnr = -$ and $arcA = +$ are the predictions

4.2.4 Case study – *E. coli* large-scale transcriptional network

The *E. coli* TRN was obtained from the RegulonDB database [SGCPG*al.*06] on November 2008. It consisted of 3250 TF-gene regulations classified according to three types: activation, repression, and context-dependent effect; we assigned +, −, and **?** signs respectively to these three types of regulation. Using the BioQuali plugin, we visualized a region where this network was inconsistent, see Fig. 4.3A. In order to correct this inconsistency we added the Sigma-gene regulations (all as positive influences), resulting in a network of 5140 regulations. This larger network was globally consistent, meaning that it may entail stable behaviors when experimental data is added to it.

We compared the globally consistent regulatory network with a small literature dataset obtained from RegulonDB in which 45 proteins/genes were carefully verified as activating (+) or repressing (−) during exponential-stationary growth shift; it was a heterogeneous dataset since the changes were reported at different time-points. This dataset of observations was initially inconsistent with the regulatory model (see Fig. 4.3B). We corrected it by adding two positive regulations from the sigma factor RpoD to $ihfA$ and $ihfB$ according to a recent publication on *E. coli* functional regulations [FGAPTQCV08a]. This network correction explained the observed repressed (−) effect of $ihfA$. The consistency of the corrected model with the small dataset reflected 498 positive and negative fluctuations in the network molecules (see Fig. 4.4).

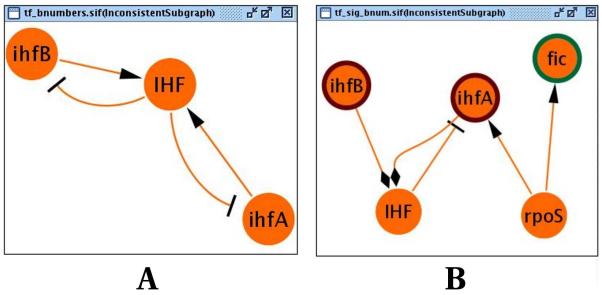


Figure 4.3: List of inconsistencies detected in the *E. coli* transcriptional network. **A.** The inconsistency appears because no possible stable behavior may be obtained using this network as $ihfA$ and $ihfB$ genes code for the protein complex IHF, which deregulates the transcription of these genes. **B.** This inconsistency was found after confronting the network with 45 literature-curated expression changes during the exponential-stationary growth shift. The nodes with red and green borders refer to $+$ and $-$ observations. The problem appears since no possible explanation exists for the negative shift observed in the $ihfA$ expression: $ihfA$ is activated by RpoS and repressed by IHF; the change in expression of RpoS was inferred to be positive (because of fic), and the change in expression of IHF was inferred to be negative (because of $ihfA$ and $ihfB$); consequently, these influences cannot explain the down-regulation of $ihfA$.

4.3 Web service

The CoryneRegNet platform [BWR+07] provides interesting information concerning the construction of bacterial regulatory networks at large-scale, and includes different types of regulatory network analyses ready to be launched on the users data. Currently, there is only one tool, COMA [BA08], that automatizes the confrontation of large-scale regulatory networks and experimental datasets. One of the motivation we had to make BioQuali reasoning available as a Web service, was to include it in a future time in the CoryneRegNet platform. It could complement the existing tool, since it is able to perform global and, thus, more complex analyses in reasonable computation time.

The BioQuali Web service makes the BioQuali functionalities accessible via SOAP requests. By using it, users can automate access to BioQuali from their own tools or databases. All requests are run on the GenOuest high performance computing facility [gen]. The output of the web service is: (1) a list of Predictions, when the network and dataset are consistent, (2) a list of local inconsistencies, algorithmically easy-to-detect network regions inconsistent with the dataset observations, (3) a list of global inconsistencies, more complex inconsistent subgraphs, (4) the core of the network, a reduction of the graph, and (5) a list of multiple inconsistencies, a reasonable-time-computing approximation of all the inconsistencies of the graph wrt a dataset.

All the documentation of the BioQuali Web service can be found at: `http://genoweb2.irisa.fr/claroline/index.php?category=NETWORKS`; also, an on-line access to it is provided at: `http://mobyle.genouest.org/cgi-bin/Mobyle/portal.py?form=bioquali`.

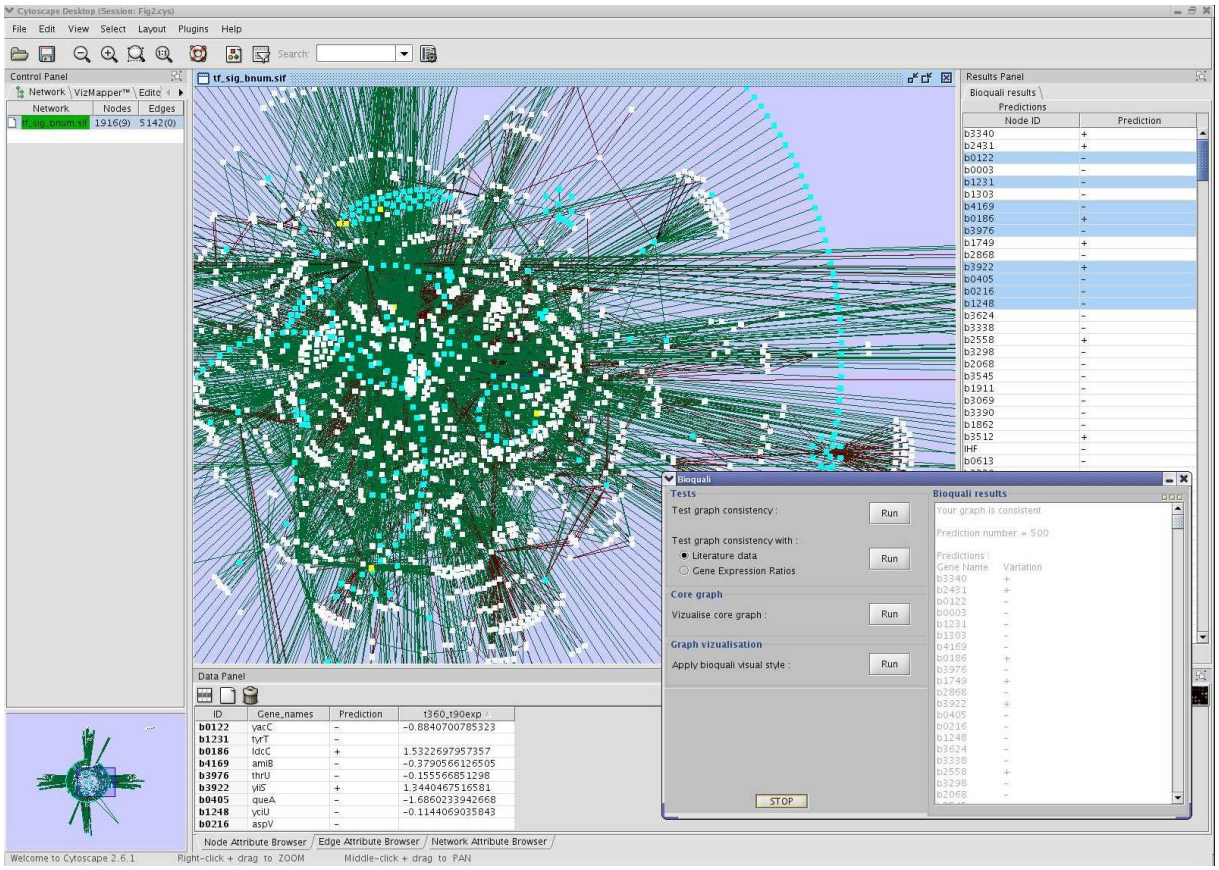


Figure 4.4: The predictions shown by the plugin in the *E. coli* network. The cyan colored nodes correspond to network products inferred as $\{+, -\}$ in order to explain the 45 gene expression observations initially provided. The plugin lists 498 predicted changes in the Results Panel; it is possible to select and visualize them in detail in the Data Panel and compare them, as shown in this image, with other experimental observations. In the bottom right corner we see the BioQuali plugin window with all the analysis options that it provides.

Chapter 5

Application to Bacterial networks

The bacterium *Escherichia coli* is one of the best-studied single-celled organisms. Its genetic regulatory network was studied by a wide community, and this information was compiled in databases such as RegulonDB. For that reason, we applied the modeling and informatic approaches proposed in Chapters 2 and 3 to this bacteria. We illustrate in this chapter the biological output obtained after applying the qualitative consistency analysis on this organism. We published part of the results presented in this chapter in [GVB+07], [VGLB+08], and [GGRS09].

5.1 Constructing the *E. coli* signed influence graph

The information on mechanisms of regulation for the *E. coli* bacteria gathered in the databases of EcoCyc [KCVGC+05] and RegulonDB [SGCPG*al.*06] is currently the largest known for a bacterial cell. The regulatory information contained in both databases is well known to be synchronized since 2005. It is presented by RegulonDB as plain text files that contain all the set of documented interactions that take place in the *E. coli* regulatory process. The information used to build the *E. coli* influence graph was obtained from two files of the RegulonDB Website:

- *RegulonDB-EcoCyc interactions.* This file contains a set of regulatory interactions at the level of transcription initiation. Each line in this file contains: *(i)* the name of the transcription factor protein, *(ii)* the regulatory gene(s), *(iii)* the regulated gene, and *(iv)* the regulatory role of the TF over the regulated gene: activator, repressor, or both. These interactions show the proteins that regulate a gene, and the gene that synthesizes the regulatory protein.

- *Genes transcribed by sigma-factors.* This file contains interactions between the sigma-factors of *E. coli* and their corresponding transcribed genes. Sigma-factors associate with the Prokaryotic RNA polymerase, and provide it the function of promoter recognition. Each σ-factor has its own specificity, allowing the initiation of transcription of different subsets of genes. In Table 5.1 we list the seven σ-factors of the *E. coli* bacteria. Expression of σ-factors in bacteria is also regulated and

some of them receive post-translational regulations. Different conditions of the cell, such as starvation, and in some cases environmental signals, such as extreme heat, can trigger the production of σ-factors.

Table 5.1: Sigma (σ-)factors present in the *E. coli* bacteria

Protein	Gene	Function
σ^{70}	*rpoD*	Transcribes most genes in growing cells
σ^{38}	*rpoS*	The starvation/stationary phase sigma-factor
σ^{28}	*rpoF*	The flagellar sigma-factor
σ^{32}	*rpoH*	The heat shock sigma-factor
σ^{24}	*rpoE*	The extracytoplasmic stress sigma-factor
σ^{54}	*rpoN*	The nitrogen-limitation sigma-factor
σ^{19}	*fecI*	The ferric citrate sigma-factor

Only the *E. coli* genes, for which exist literature evidence confirming their interaction with other genes, are present in the interaction files of RegulonDB. In the following sections we show how we built the influence graph for the *E. coli* bacteria using the information gathered in RegulonDB.

5.1.1 *E. coli* influence graph - only transcriptional regulations

From the file *RegulonDB-EcoCyc interactions* we built the influence graph of the *E. coli* network as the set of regulations of the form '$A \rightarrow B$ *sign*'. The value *sign* is the role of the TF regulating the gene. Its value is in $\{+, -, \mathbf{?}\}$ meaning: activator, repressor, or dual or complex effect. A and B can be considered as genes or proteins, depending on the following situations:

- The relation '$genA \rightarrow genB$ *sign*' was created when protein A, synthesized by $genA$, was the transcription factor regulating $genB$. The regulatory role of the transcription factor is given by *sign*, which value is in $\{+, -, \mathbf{?}\}$ (see Fig. 5.1A).
- The relation '$TF \rightarrow genB$ *sign*' was created when TF was an heterodimer protein-complex regulating $genB$. The regulatory role of TF is given by *sign*, which value is in $\{+, -, \mathbf{?}\}$ (see Fig. 5.1B). There were four protein-complexes in the *E. coli* TRN obtained from RegulonDB: IHF, HU, RcsB, and GatR.
- The relation '$genA \rightarrow TF$ +' was created when TF was an heterodimer protein-complex synthesized by $genA$ (see Fig. 5.1B).

The influence graph for *E. coli* is only built from transcriptional information. The metabolites and signals were not taken into consideration for the construction of this graph. The first resultant influence graph, compiled on March 2006, had 1258 nodes (genes and protein-complexes) and 2526 interactions. In Figs. 5.2 and 5.3 we illustrate a partial and global view of this graph.

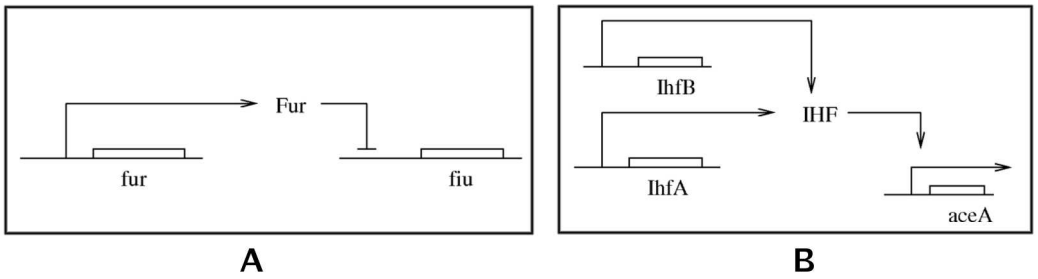


Figure 5.1: Influence graph representing the *E. coli* genetic interactions. **A.** Negative regulation (repression) of gene fiu by the transcription factor Fur represented as '$fur \rightarrow fiu -$' in the influence graph. **B.** Biological interaction of genes $ihfA$ and $ihfB$ forming the protein-complex IHF represented as '$ihfA \rightarrow IHF +$' and '$ihfB \rightarrow IHF +$'. Positive regulation of gene $aceA$ by the protein complex IHF represented by '$IHF \rightarrow aceA +$' in the influence graph.

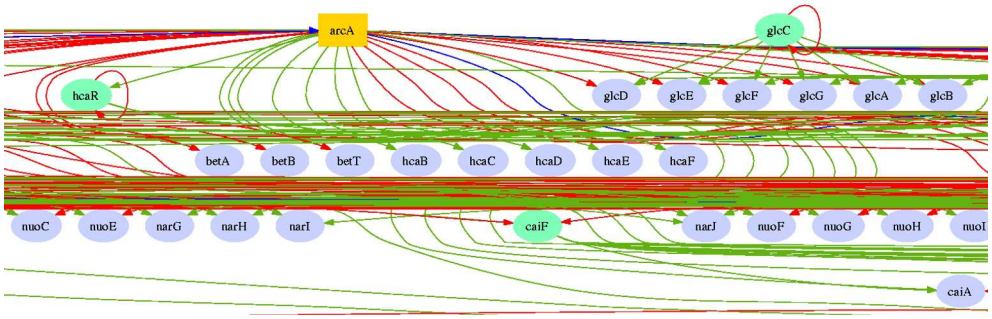


Figure 5.2: Partial view of the *E. coli* influence graph associated to its TRN. Regulated genes are shown as blue ovals, TFs as green ovals, and global TFs as yellow rectangles. Green (resp. red) edges represent activations (resp. repressions); blue arrows represent dual or complex interactions.

The *E. coli* TRN is well known for its hierarchical structure [MKD+04, MACV03]. In Fig. 5.4 we show a comparison between the number of predecessors and successors by gene that appear in this network. 87% of the total number of nodes is regulated by the 23% that remains, that is, the network holds a power-law distribution. Among this 23% there are seven regulatory proteins that directly regulate more than 80 genes each one. These proteins are known as global factors and are: CRP, FNR, IHF, FIS, ArcA, NarL, and Lrp. Another characteristic of this network is that it contains certain network motifs [SOMMA02], such as feed-forward loops and regulons.

Notice that the RegulonDB database is constantly being updated, thus, new regulatory interactions appear each year. We have presented the 2006 version of the *E. coli* TRN. We have, however, built two more *E. coli* influence graphs following the same steps described in this section. These graphs were built using the available information

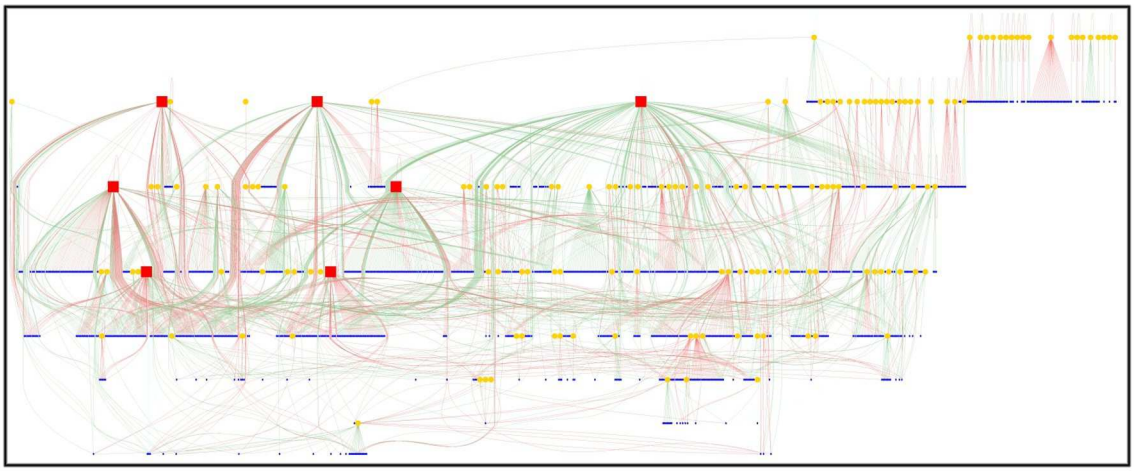


Figure 5.3: Global view of a the *E. coli* influence graph. This network was obtained from RegulonDB on March 2006. It consisted of 1258 nodes and 2526 edges. Regulated genes are shown as blue ovals, TFs as yellow ovals, and global TFs as red squares. Green (resp. red) edges represent activations (resp. repressions).

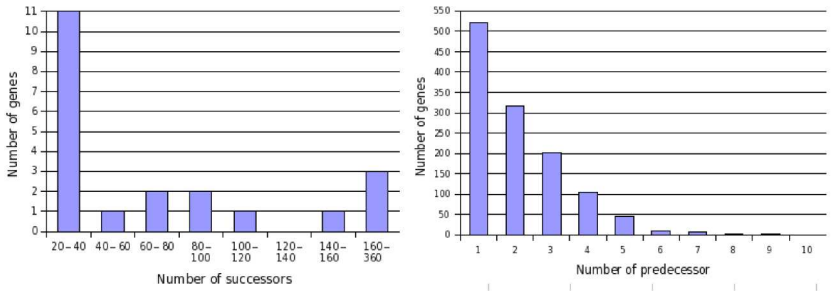


Figure 5.4: Charts illustrating the distribution of number of genes in the *E. coli* TRN (Y axis), controlling (left) or being controlled (right) by groups of genes of different sizes (X axis).

of RegulonDB in 2007 and 2008. We can see their number of nodes and edges in Table 5.2. We will use these three graphs in the different analyses presented in this chapter.

5.1.2 Adding sigma factors to obtain self-consistency

Using the analysis described in Section 2.2.3 we built the qualitative system of constraints for the *E. coli* influence graph (compiled in 2006). Afterwards, we used the Bioquali Python package described in Section 3.1 to decide the consistency of this system of constraints. The system was found *self-inconsistent*. Thus, we applied the Bioquali functionality to isolate a minimal inconsistent subgraph (see Figure 5.5). A careful reading of the available literature led us to consider the regulations involving sigma-factors which were initially absent in the influence graph. These new regulations were

taken from the *Genes transcribed by sigma-factors* file in RegulonDB. They were added in the influence graph as a list of positive regulations of the form 'σ-factor $\rightarrow$ *gene* +', since sigma-factors are well known to enhance the transcription of genes. Once they were added to complete the network, we obtained a network of 3802 interactions and 1529 components (genes, protein-complexes, and sigma-factors). This final network was found to be self-consistent.

We also computed the self-consistency of the influence graphs compiled in 2007 and 2008 composed only of TF-gene regulations. As result, we obtained that they were self-inconsistent. Thus, they were extended using the σ-factors regulations currently available on that years. Two larger influence graphs were compiled; both were self-consistent. See their number of nodes and edges in Table 5.2.

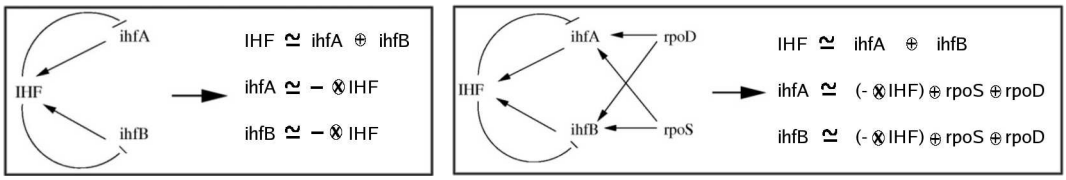


Figure 5.5: (Left) A minimal inconsistent subgraph, isolated from the *E. coli* influence graph using the Bioquali package functionalities. (Right) Correction proposed after careful reading of the available literature on *ihfA* and *ihfB* regulation.

Table 5.2: Sizes of three influence graphs built using the RegulonDB information available in 2006, 2007, and 2008. Graphs composed only of TF-gene regulations were self-inconsistent; while those composed of TF- and σ-regulations were self-consistent.

Year	TF-gene regulations	TF-gene and σ-gene regulations
2006	1258 nodes - 2526 edges	1529 nodes - 3802 edges
2007	1415 nodes - 2899 edges	1763 nodes - 4491 edges
2008	1499 nodes - 3250 edges	1915 nodes - 5140 edges

Biological interpretation of the results 5.1. *Transcriptional regulations are not enough to explain by themselves the variations of any product of the system. The sigma-factors interactions cannot be omitted in the construction process of the final model for E. coli.*

5.1.3 Core of an influence graph

Obviously, not all interactions play the same role in the network. The *core* is a subnetwork that naturally appears for computational purpose and plays an important role in the system. It consists of all oriented loops and of all oriented chains leading to loops. All oriented chains leaving the core without returning are discarded when reducing the

network to its core. Acyclic graphs and, in particular, trees have no core. The main property of the core is that if a system of qualitative equations has no solution, neither has the reduced system built from its core. Hence it corresponds to the most difficult part of the constraints to solve. It is obtained by reduction techniques explained in Chapter 3. As an example, the core of the *E. coli* influence graph obtained from RegulonDB in 2006 and composed of TF- and σ-gene regulations had only 28 nodes and 57 edges (see Fig. 5.6).

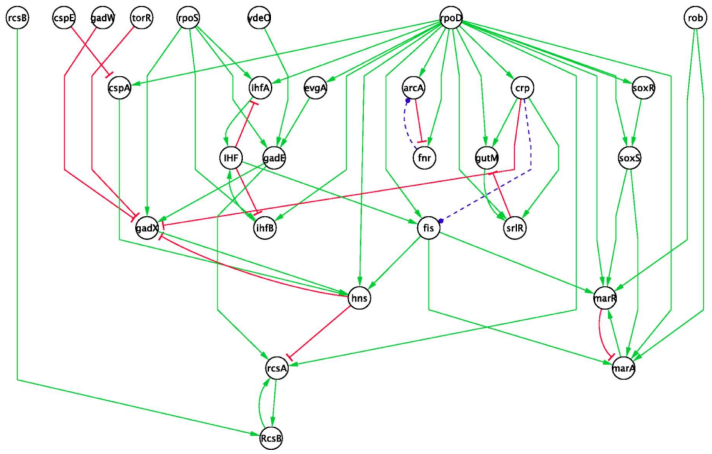


Figure 5.6: Core of the *E. coli* influence graph obtained from RegulonDB in 2006, composed of TF- plus σ-gene regulations.

5.2 Network consistency wrt a single dataset

5.2.1 Dataset used in the consistency analysis

Our goal was to confront the self-consistent influence graph of *E. coli* to a stress-perturbation experimental condition. Thus, we compared two conditions of the cell at steady state: *(i)* exponential growth, and *(ii)* stationary growth. These conditions were compared to study the exponential-stationary growth shift in the *E. coli* bacteria. The dataset, composed of differential observations of some *E. coli* genes affected during this growth shift, was collected from the literature based on initial information provided in RegulonDB. These observations were supported by a precise evidence and literature citation, and were classified as induced or repressed.

We processed this information in order to assign qualitative $\{+, -\}$ changes to some nodes of the influence graph. Thus, we compiled a set of observations of the form '$gene = sign$', where $sign$ was set to '$+$' or '$-$' depending if the gene was expressed or, respectively, repressed under this condition. In Table 5.3 we show the processed dataset corresponding to the qualitative changes of 45 *E. coli* genes during the exponential-stationary growth shift.

Table 5.3: 45 gene expression changes observed in *E. coli* during the exponential to stationary growth shift. This dataset was collected from the literature, based on initial information provided in RegulonDB. The Locus column corresponds to the Blattner numbers [BrPB+97]. Notice that for $ihfA$ RegulonDB proposes a '−' sign. We corrected this sign as it is incompatible with both our consistency analysis and the literature provided (see next section for details).

Locus	Gene	Sign	Locus	Gene	Sign	Locus	Gene	Sign	Locus	Gene	Sign	Locus	Gene	Sign
b0464	*acrR*	+	b0978	*appC*	+	b1241	*adhE*	+	b1814	*sdaA*	−	b2663	*gabP*	+
b2163	*yeiL*	+	b0979	*appB*	+	b1732	*katE*	+	b1197	*treA*	+	**b1712**	***ihfA***	+
b0564	*appY*	+	b3855	*rrfA*	−	b4149	*blc*	+	b4233	*mpl*	+	b0912	*ihfB*	−
b2679	*proX*	+	b3701	*dnaN*	+	b3500	*gor*	+	b2552	*hmp*	+	b1897	*otsB*	+
b4396	*rob*	+	b0054	*imp*	+	b1482	*osmC*	+	b3517	*gadA*	+	b0435	*bolA*	+
b1276	*acnA*	+	b2579	*yfiD*	+	b1283	*osmB*	+	b1493	*gadB*	+	b0972	*hyaA*	+
b4376	*osmY*	+	b3361	*fic*	+	b1739	*osmE*	+	b1492	*gadC*	+	b3544	*dppA*	+
b0463	*acrA*	+	b1272	*sohB*	−	b3863	*polA*	+	b0899	*lrp*	+	b3700	*recF*	+
b1896	*otsA*	+	b4111	*proP*	+	b0880	*cspD*	+	b2535	*csiE*	+	b1237	*hns*	+

5.2.2 Feasibility of the consistency-check

In 2006, a first consistency analysis was performed using the 2006 version of the *E. coli* TRN (composed of TF- and σ-gene regulations) and the exponential-stationary dataset. Our goal was to test the feasibility of our approach on a large-scale regulatory network. The results obtained were published in [GVB+07]. We summarize the main results in the following lines:

1. **Consistency answer**. The qualitative changes reported by the exponential-stationary dataset were inconsistent with the *E. coli* influence graph.

2. **Diagnose of the inconsistency**. The dataset of observations proposed by RegulonDB assigned the *ihfA* gene a '−' change during this condition. Thus, in this first analysis we detected an inconsistency (see explanation in Fig. 5.7).

   We corrected the $ihfA$ observed value setting its variation to '+', as shown in Table 5.3. This correction was done on the basis of the studies reported in [AAIN+91, AGS+94], which agree that the transcription of $ihfA$ increases during stationary phase. After correcting this dataset observation, the *E. coli* influence graph was consistent with the exponential-stationary dataset.

3. **Predictions**. Once the *E. coli* influence graph was consistent with the dataset, we explored the solutions of the qualitative system. There were about $2,66 \cdot 10^{16}$ solutions consistent with the observations of the exponential-stationary growth shift. In them, 381 variables of the system were always fixed to the same value, and thus corresponded to the predictions.

Biological interpretation of the results 5.2. *The E. coli influence graph composed of TF- plus σ-gene regulations obtained from RegulonDB in 2006 was inconsistent with the* 'ihfA $= -$' *observation provided in the dataset of the transition from exponential*

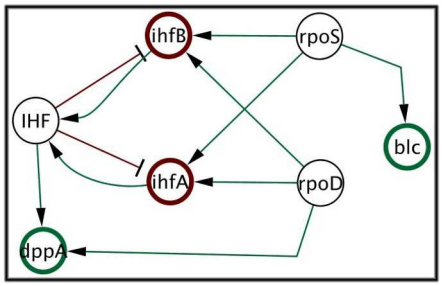


Figure 5.7: Inconsistent graph obtained with the data available in RegulonDB in 2006. Nodes with a green (resp. red) border correspond to '+' (resp. '−') observations in the exponential-stationary growth shift dataset. The green (resp. red) arrows correspond to activations (resp. inhibitions). Explanation: (1) Since *ihfA* and *ihfB* are observed to be down-regulated (−), then IHF will be fixed to '−'. (2) If this is the case, then the *dppA* gene '+' observation must be explained by RpoD's change, that will be fixed to '+'. (3) The inconsistency appears because the '−' observation of *ihfA* is not explained neither by RpoD (set to '+' in step 2), nor by RpoS (set to '+' because of *blc*'s observation).

to stationary growth of the cell. We corrected the observed value for the ihfA gene in agreement with the studies reported in [AAIN+91, AGS+94].

5.2.3 Validation of the consistency-check

In 2007, we compiled a second influence graph for the *E. coli* TRN using the regulatory information available in RegulonDB to that date. The interactions in the influence graph represented transcriptional, complex formation, and sigma-gene regulations. The products of the network were genes, active proteins, and protein-complexes. The *E. coli* influence graph was composed of 1763 products and 4491 interactions.

Our objective was to validate our approach by comparing our computational predictions with respect to an independent dataset of transcriptional data. Thus, we tested the consistency of this graph using the exponential-stationary growth shift condition (*cf.* Table 5.3) in order to evaluate the biological impact of our results. The exponential-stationary dataset had the corrected observation for *ihfA* ('*ihfA* = +'). We published this work in [GGRS09]; in this section we describe our main results.

Consistency test and diagnose

We started our analysis by constructing a system of qualitative constraints as explained in Section 2.2.3 from the *E. coli* influence graph (obtained in 2007) and the exponential-stationary growth shift dataset. Then, we computed its consistency by using the Bioquali Python package. The first result obtained was an inconsistency between the network and the experimental dataset. This inconsistency corresponded to a newly added transcriptional interaction ('`hns -> appY -`') that appeared in RegulonDB in 2007. The inconsistent region, highlighted by the methods in Bioquali, is shown in Fig. 5.8A.

It represents the inhibition of transcription of the *appY* gene by the H-NS protein. No other transcriptional regulation of the *appY* gene was found in the RegulonDB database. These products (*appY*, *hns*, and therefore H-NS) are, however, shown to increase their levels in the exponential-stationary growth shift of the cell [DSB93, ASOB96]. Hence, the source of the conflict may be in the model.

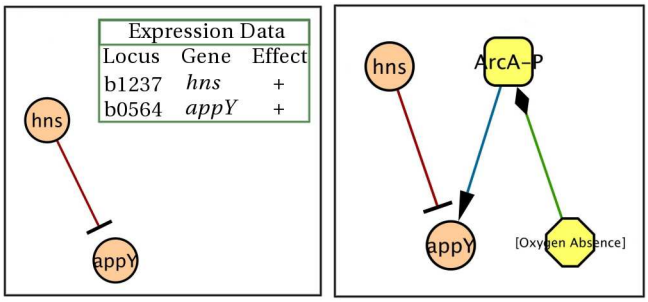


Figure 5.8: Diagnosis when an inconsistency between the model and data is found. **A**. The inhibition of gene *appY* by the *hns* product causes an inconsistency with the expression data related to the exponential-stationary growth shift. **B**. Correction of the inconsistency by adding a positive regulation from ArcA-P (phosphorylated protein ArcA) to *appY*; this regulation occurs in the absence of oxygen.

Searching in the primary literature we found that the *appY* gene is induced during entry into stationary phase, and that during oxygen-limiting conditions the stationary-phase induction is partially dependent on ArcA [BA96]. The protein ArcA is activated via phosphorylation by the ArcB sensor under conditions of reduced respiration [ICF+90]. The signal which leads to the activation of ArcA during entry into stationary phase may be the deprivation of oxygen caused by an increase in cell density [BA96]. Based on these studies we corrected our influence graph adding new interactions (see Fig. 5.8B), obtaining an influence graph consistent with the experimental data.

Biological interpretation of the results 5.3. *The E. coli influence graph, composed of TF- plus σ-gene regulations, obtained from RegulonDB in 2007, was inconsistent with the following observations: '*hns = +*' and '*appY = +*' obtained from the exponential-stationary growth shift dataset. These observations are correctly validated in the literature. Thus, we added one regulation into the model '*`ArcA-P -> appY`*' in agreement with the studies reported in [BA96].*

Computational predictions and validation

Once our network was consistent with the expression data provided for the exponential-stationary growth shift, we generated the computational predictions. From the 45 gene expression changes reported by the dataset, we obtained 20 predictions (1% of the total network nodes). Our system of qualitative constraints was built by using the generic qualitative function *GEN* (*cf.* Equation 2.2) to model the regulations arriving to a

node in the network. In Section 2.3.1 we proposed a new qualitative constraint to model precisely the protein-complex formation (*cf.* Equation 2.5). By applying this rule to the IHF protein-complex, our prediction results changed considerably.

The new set of predictions obtained was composed of 502 changes in network components (30% of the total network nodes). We characterized these predictions into 12 functional groups (see Table 5.4) using the *DAVIS* software [DSH+03]. To validate our computational predictions, we obtained from the *Many Microbe Microarray Database* [FDF+07] a dataset of differentially expressed genes after 720 minutes of growth (stationary phase) in a rich medium [AHL+03]. This dataset was compared to our predictions (see Figs. 5.9 and 5.10).

Table 5.4: 12 functional groups into which the computational predicted network products were classified.

ID	Description
A	Amino acids metabolism and biosynthesis
C	Carbohydrates metabolism and biosynthesis
E	Energy metabolism
G	Glucose catabolism
L	Lipid metabolism and biosynthesis
N	Nucleic acids metabolism

ID	Description
R	Regulatory function
S	Cell structure
SI	Signal peptides
T	Transport
V	Vitamin metabolism and biosynthesis
U	Unassigned

Expression profiling of the microarray dataset identified 926 genes that changed significantly (2-fold) in transcription in response to the growth shift from exponential to stationary phase. The 502 products, computationally predicted, could be classified into four categories: 130 agreed with significant expression changes, 32 had a predicted expression change in a direction opposite to that of the experimental data, 306 had a predicted expression change that was not found to be statistically significant in the experimental data, and for 34 products there was no expression data available (some products of the network were protein complexes and could not be compared to mRNA expression). Thus, of the 162 (=130+32) significant differentially expressed genes that could be compared between the computational predictions and the experiment, 130 (or 80% consensus) agreed. Only the 32% (coverage) of our predictions could be compared with 2-fold expression changes. Therefore, we performed the same analysis choosing different thresholds (1.5-fold, 0 fold). In this way, new consensus and coverage percentages were calculated showing that the higher the threshold is, the better is the consensus and – as expected – the worse is the coverage of the predicted data (Fig. 5.11).

Biological interpretation of the results 5.4. *The percentage of gene expression predictions obtained when confronting the E. coli influence graph wrt 45 observations of the exponential-stationary growth shift increased in* 30% *when modeling the IHF complex formation using quantitative data related to the expressions of ihfA and ihfB.*

Biological interpretation of the results 5.5. *80% of our computational gene expression predictions were in agreement with the mRNA measurements of an independent microarray study on the exponential-stationary growth shift.*

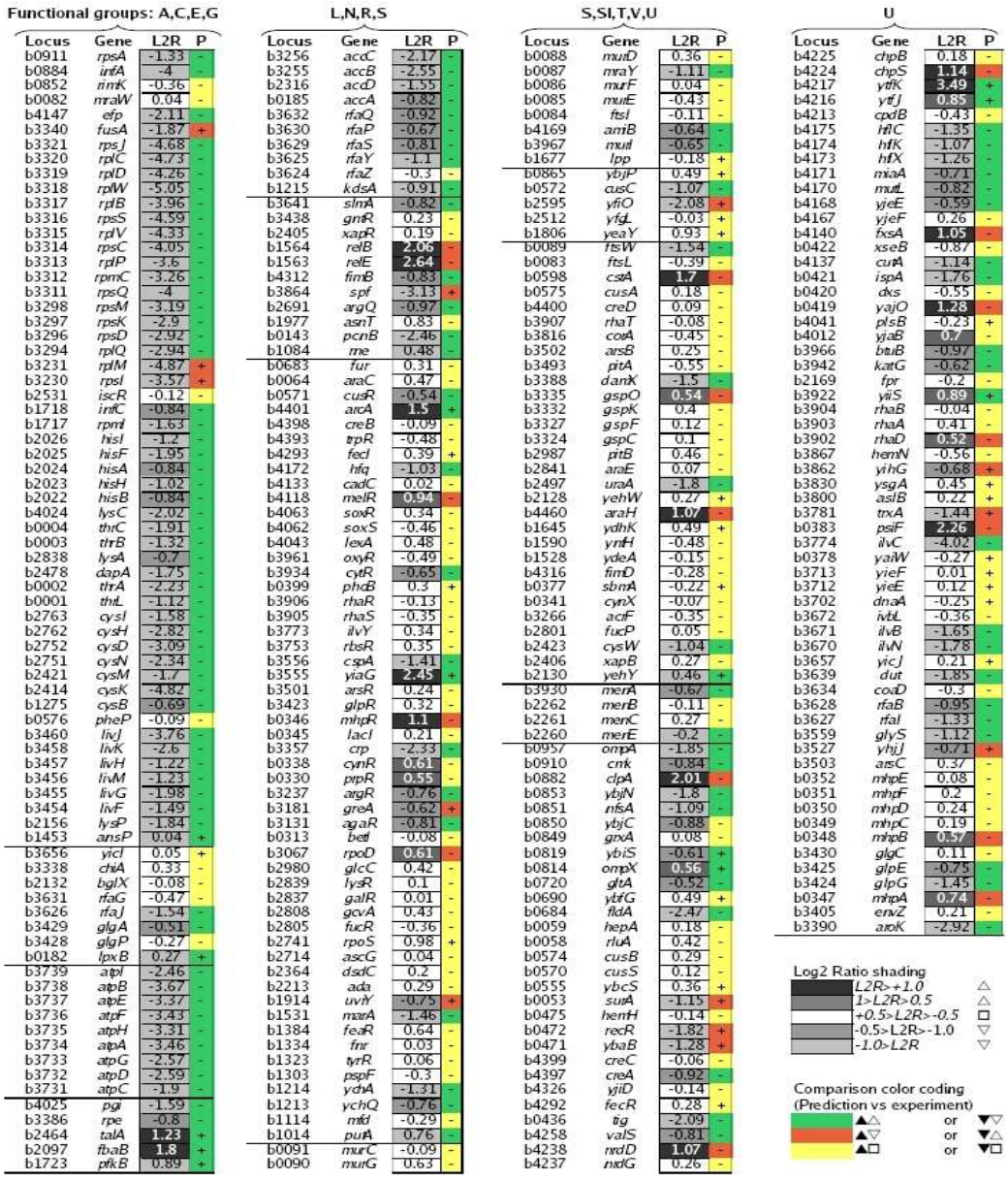


Figure 5.9: Table of microarray-observed vs. predicted gene-expression responses in the *E. coli* network under the exponential-stationary growth shift condition. The locus numbers, gene names, and the log_2 ratio (L2R) of gene expression (exponential to stationary) are shown for some of the 502 predicted expression changes (+,−). Genes were divided in 12 functional groups (Table 5.4). The L2R is shaded depending on the magnitude of the expression shift. Filled and open symbols indicate computational predictions and experimental data, respectively, squares indicate no change in gene expression, and triangles indicate a change in expression, as well as the direction of change (up-regulated or down-regulated).

Threshold	2-fold	1.5-fold
Exp. total	926	1757
Pred. total	502	502
▲△ or ▼▽	130	195
▲▽ or ▼△	32	62
▲□ or ▼□	306	211
No possible comparison		
▲∗ or ▼∗	34	34
∗△ or ∗▽	458	1289

Figure 5.10: Comparison between predicted and microarray-observed expression changes. An ∗ symbol indicates either that our model did not predict a gene expression or that no expression data related to a gene in our model was found. Filled and open symbols indicate computational predictions and experimental data, respectively, squares indicate no change in gene expression, and triangles indicate a change in expression, as well as the direction of change (up-regulated or down-regulated).

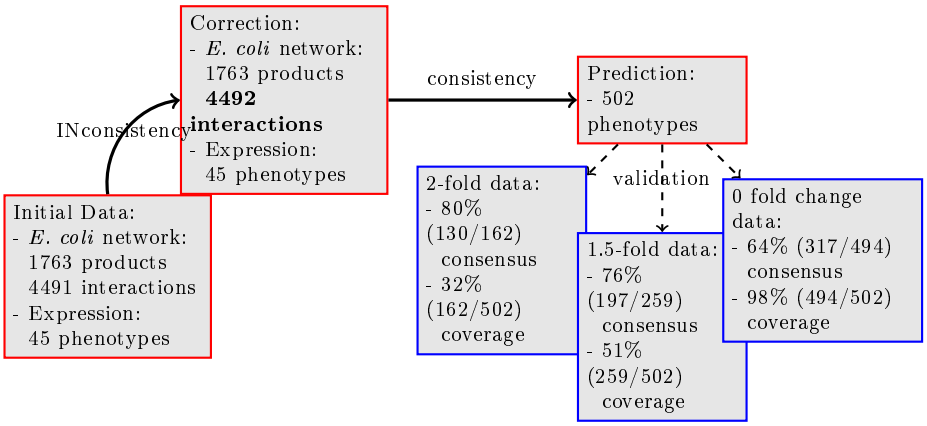


Figure 5.11: Results of the consistency check process applied to the *E. coli* transcriptional network using 45 phenotypes related to the exponential-stationary growth shift. We validated our computational predictions using the observations in a microarray dataset filtered with three thresholds. Consensus refers to the percentage of validated model predictions, and coverage indicates the percentage of compared predictions.

5.2.4 Discussion

We illustrated in the previous sections the results obtained with the consistency-check process applied to the large-scale *E. coli* TRN. This analysis searched: *(i)* to confront existing biological data, *(ii)* to highlight the inconsistencies between a regulatory network and a experimental dataset, and *(iii)* to predict non-observed qualitative changes in some network molecules, which explain the dataset observations.

By applying our analysis, we proposed corrections to the regulatory information

given by RegulonDB. First, we highlighted an inconsistency in the dataset (change of *ihfA*). Second, we highlighted a missing regulation in the RegulonDB *E. coli* TRN. Both corrections were made based on studies reported in the literature.

In addition, we evaluated the validity of our computational predictions by comparing them to an independent microarray study. The microarray dataset corresponded also to the exponential-stationary growth shift in *E. coli* cells. In this condition appear three important phases: exponential, early stationary-phase, and late stationary-phase. During the shift among these phases many molecules in the cell change their behavior considerably. In our study we only compared the first and last phases. We believe that these two conditions represent instants in the cell where the genes and proteins do not significantly change their concentration. Nevertheless, the 45 observations may correspond to slightly different time points of the exponential phase and thus induce divergences in our computational predictions. In spite of this, a high percentage of our predictions (80%) was validated when compared with the microarray data. This percentage is comparable to the one obtained by other methods working on a more complex and precise *E. coli* regulatory model [CKR$^+$04, CP02, EP00].

After the comparison of our computational predictions with microarray measurements, some disagreements were detected. A reasonable explanation for these divergences resides on our previously made assumption that some level of correlation exists between the transcription factor protein and the target gene expression without considering in detail the post-transcriptional effects. This problem was also reported in [HCP03]. An example of this case of disagreement is illustrated in Fig. 5.12. This type of errors in our predictions can be useful to complete the regulatory model with post-transcriptional regulations.

We have built the qualitative system of constraints to model the *E. coli* regulatory data, using the generic qualitative function. We also used quantitative data of *ihfA* and *ihfB* expression to model the IHF complex-formation. Based on this information, we applied the consistency-check process described in Section 2.2.3. Our analyses were computed using the Bioquali package; specifically, by running the program described in Algorithm 1. The computational time of the consistency-check analysis took less than one minute. Our results not only proved the feasibility of a global and automatic analysis of large-scale TRNs. They also reflected important global configurations in a regulatory network that can be practically used to diagnose models and predict expected behaviors of the system.

5.3 Network consistency wrt wide-genome datasets

In the previous section, we described the results obtained when a large-scale TRN was confronted to a small dataset (45 observations) obtained from the literature. In that case it was feasible to correct the punctual reported inconsistencies by preforming an extensive search in the literature.

The regulatory information gathered by RegulonDB for the *E. coli* bacteria is far

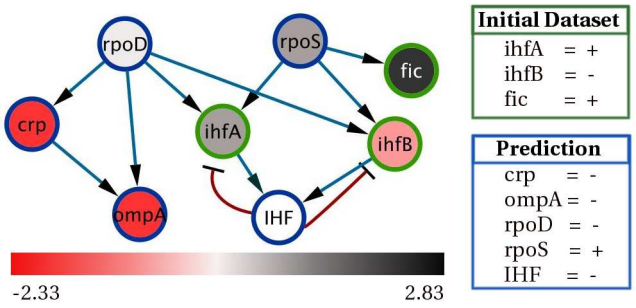


Figure 5.12: Consistency check process for 8 products of the *E. coli* regulatory network under the exponential-stationary growth shift. All transcriptional influences that each product receives appear in the network. The gray-red intensity of each product reflects the experimentally observed change in mRNA expression (log_2 ratio) during the studied condition. Products with a green border refer to those present in the initial dataset obtained from the literature, whereas products with a blue border refer to our computational predictions. Experimentally observed mRNA-expression changes agree with the computational predictions except for RpoD, where the predicted changes correspond to variations on the *active protein* and cannot be observed on mRNA expression levels. The decrease of the active protein RpoD was reported in [JI98].

from being complete [EBK+08]. Thus, when testing its consistency wrt larger experimental datasets, the number of inconsistencies will naturally increase. The manual correction of multiple inconsistencies is unbearable. Therefore, another type of diagnosis in the consistency analysis was proposed. In Chapter 3 we presented a couple of programs, implemented using either TDDs or ASP that performed automatic corrections when a network and dataset hold many inconsistencies. In this section we present and validate the results obtained by using these programs with the *E. coli* TRN and genome-wide microarray measurements. For this purpose, we used the self-consistent *E. coli* influence graph built in 2008, which was composed of 1915 nodes and 5140 edges. We submitted, in collaboration with the team leaded by T. Schaub in the Potsdam University, part of the work presented in this section to the "Twelfth International Conference on the Principles of Knowledge Representation and Reasoning", to be held in Toronto, Canada, May 9-13, 2010.

5.3.1 Genome-wide datasets used in the consistency analysis

We collected two genome-scale datasets by confronting two different *E. coli* Affymetrix expression compendium [FHT+07]. These datasets corresponded to two different conditions:

(A) *Stationary vs. exponential growth shift.* This dataset was composed of 4298 $\{+, -\}$ gene expression changes, obtained by comparing the stationary growth phase with the early-log growth phase in the *E. coli* K12 strain [BMA+07]. From this large dataset we extracted two subsets of observations. The first one corresponded to significant 3-fold changes and consisted of 255 observations. The second one corresponded to 2-fold changes and consisted of 855 observations.

(B) *Heat shock vs. control.* This dataset was composed of 4298 $\{+, -\}$ gene expression changes, obtained by comparing *E. coli* wild-type cells under heat-shock stress with non-stressed cells [AHL$^+$03]. From this large dataset we extracted a subset of 2-fold significant observations. It consisted of 879 observations.

5.3.2 Finding all inconsistent subgraphs in the network

In Chapter 3 we presented two programs to find the inconsistent subgraphs in the network. The first one, based on TDDs and described in Algorithm 2, does not find exactly all inconsistent graphs, but approximates them. The second one, based on ASP and proposed in [GST$^+$08], finds exactly all the inconsistent subgraphs. While the second approach produces better results, the authors are still investigating how to increase its computation time when confronted with large-scale networks.

The difference between the ASP method and the one using TDDs is that with ASP we completely explore all the possible inconsistencies. Recall that the program based on TDDs (*cf.* Alg. 2) cannot afford exploring all inconsistencies. Instead, as soon as it finds an inconsistent subgraph it deletes it from the graph. This deletion implies loosing edges in the network that could lead us to another inconsistency.

We applied both programs to the *E. coli* influence graph obtained in 2008 and composed of TF- and σ-gene regulations. The results are detailed below.

Approximating inconsistent subgraphs - TDDs

We checked the consistency of the *E. coli* influence graph wrt 3-fold changes of dataset A (*cf.* Section 5.3.1). The dataset composed of 3-fold changes had 255 observations. Applying the program described in Algorithm 2 (*cf.* Section 3.1.2.1) we obtained 16 inconsistent subgraphs. These subgraphs were merged into a single graph of 44 nodes and 40 edges. In Fig. 5.13 we see all the inconsistencies found; we show them wrt the whole dataset of gene expression measurements. The computation time of this analysis was of 140 s.

Finding exactly all inconsistent subgraphs - ASP

We applied the ASP program proposed in [GST$^+$08] to compute all the inconsistencies between the *E. coli* influence graph and the dataset of 255 observations. While using the same data as the previous method, the obtained results were considerably different (see Fig. 5.14). As expected, by computing exactly all the inconsistencies we built a larger graph of 134 nodes and 219 edges.

The graph, built from all the inconsistencies, obtained using ASP is more connected than the graph shown in Fig. 5.13. The execution of the ASP encodings, however, are costly in computation time. By waiting long enough we could obtain the results shown in Fig. 5.14. This difficulty needs to be improved in order to diffuse this method via bioinformatic tools.

The graph shown in Fig. 5.14 does not mean that all its edges and observations are incorrect. It provides a complete view of the inconsistencies in order to propose

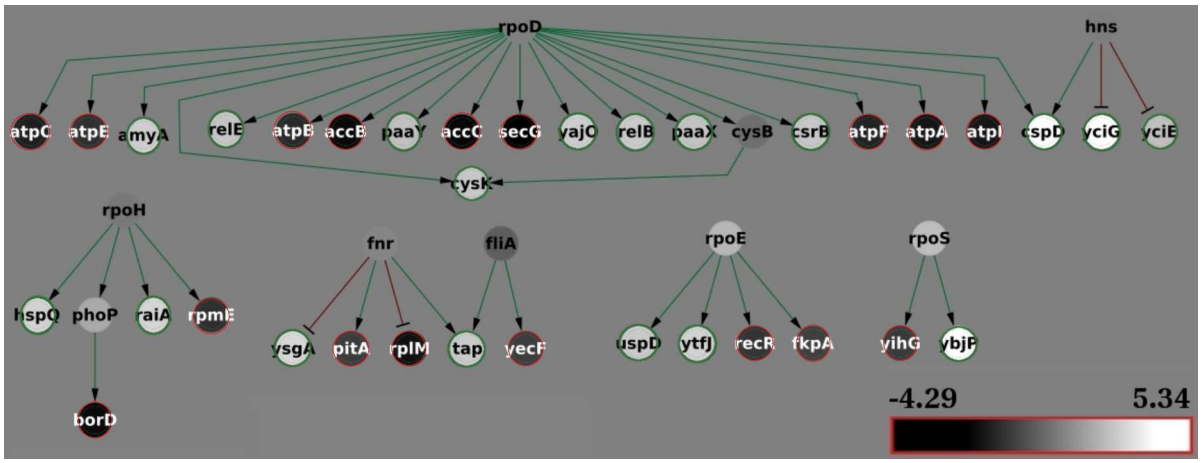


Figure 5.13: Inconsistent graphs detected when confronting the large-scale *E. coli* TRN with 255 (3-fold significant) gene expression changes of the exponential-stationary growth shift. The green/red border color of the nodes represents the $+/-$ changes in the 3-fold significant dataset. The gene expression changes reported by the whole dataset (4298 observations) appear colored in a black-white scale, the background color represents no change.

intelligent automatic corrections. In Section 3.2.3 we proposed an ASP program to automatically correct an inconsistent graph. This program finds out the minimal changes to make in the network and/or dataset in order to reestablish their consistency. We applied this method to the *E. coli* TRN and the 255 exponential-stationary microarray observations. We obtained that to reestablish the consistency between them, we needed to shift the sign of 11 network products (see Table 5.5). These products appear colored in yellow in Fig. 5.14.

Table 5.5: 11 *E. coli* products, which observation change (reported in the 255 microarray observations) needs to be changed in order to reestablish the consistency of the whole *E. coli* network and dataset.

Locus	Gene	Observed sign
b4408	*csrB*	+
b1400	*paaY*	+
b1927	*amyA*	+
b1563	*relE*	+
b1564	*relB*	+
b3862	*yihG*	−

Locus	Gene	Observed sign
b1399	*paaX*	+
b3936	*rpmE*	−
b2414	*cysK*	+
b0419	*yajO*	+
b0880	*cspD*	+

This is a very interesting result that can be applied to complete a regulatory network. Indeed, the proposed changes in the signs of the 11 observations reported in Table 5.5 may mean that these products receive an unknown regulation. In many cases they appear to be co-regulated with other genes, which change in expression corresponds to that of the regulatory TF.

Let us take as an example the case of the *yihG* gene. This gene forms part of a regulon regulated by RpoS. RpoS is well known to be the σ-factor that up-regulates

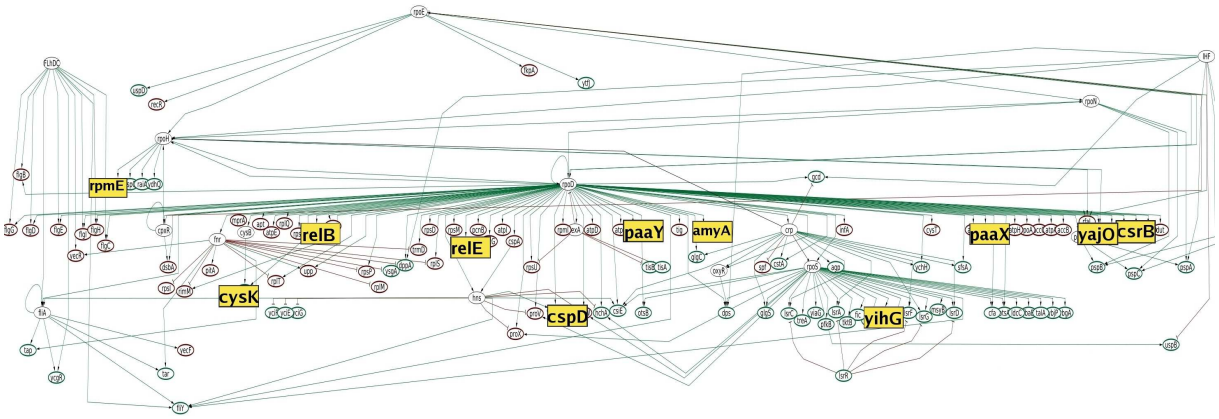


Figure 5.14: Exactly all the inconsistent graphs detected when confronting the large-scale *E. coli* TRN with 255 (3-fold significant) gene expression changes of the exponential-stationary growth shift.

during the exponential-stationary transition. All the genes in the RpoS regulon are observed in the microarray to be up-regulated except for *yihG*. Instead of questioning the observed sign of all the rest of genes targeted only by RpoS, by minimizing the repairs we highlight the sign of only one gene. This automatic discovery of inconsistencies could thus appear to be significant if we consider that both the model and dataset should express topological and gene expression coherency.

Biological interpretation of the results 5.6. *If we consider that the topological structure of the E. coli influence graph is highly consistent with the 3-fold significant mRNA expression changes reported by the exponential-stationary growth shift microarray dataset, then the 11 products reported in Table 5.5 could be potential candidates for further TFBMs exploration or literature search in order to complete the regulatory knowledge in databases such as RegulonDB.*

5.3.3 Prediction after automatic correction of inconsistencies

In Section 5.3.2 we presented two ways to automatically correct an inconsistency. The first one, implemented on TDDs, consists of deleting iteratively the edges of the inconsistent graph found. The second one, implemented on ASP, finds the minimum of all possible corrections that can be applied either to the network or dataset. By correcting our network/dataset using one of these methods, we end with a consistent regulatory network wrt a *large* dataset. From a consistent qualitative system of constraints it is possible to generate predictions of new $\{+, -\}$ changes of some network products.

Our objective now is to validate the automatic correction approaches. Recall that in the previous section we reported that even if the global set of inconsistencies between the network and dataset was large, by correcting only 11 observed network products we could conciliate the whole data. Now, we search to validate this result by comparing

the obtained predictions when minimizing the inconsistencies, with the measurements reported in the dataset.

On that account, we validated the accuracy of the predictions generated after applying the two methods proposed in Section 5.3.2. For this we used two larger datasets. The first one consisted of 2-fold significant changes of dataset A (*cf.* 5.3.1), it had 855 observations; we will refer to it from now on as *dataset A2*. The second, consisted of 2-fold significant changes of dataset B (*cf.* 5.3.1), it had 879 observations; we will refer to it as *dataset B*. As before, we will use the large-scale *E. coli* influence graph extracted in 2008 and composed of TF- and σ-gene regulations.

In order to validate the predictions generated after automatically correcting the network and/or dataset, we performed the steps described in Algorithm 7. The function `extractSample(`p, μ`)` randomly chooses observations from μ, and outputs a subset of observations of size $p * |\mu|$. The function `agreement(`$pred, X$`)` compares two datasets of observations and outputs the percentage of agreement, *i.e.* in how many cases variations available in the whole dataset (X) were predicted.

Algorithm 7 Prediction validation

Require: $\mathcal{N} = (V, E, \sigma)$; $\mu = \{(n, s) \mid n \in V, s \in \{+, -\}\}$; p, a fraction of the dataset used to test the consistency
Ensure: *avgAcc*, the average of the accuracy of the predictions obtained in all tests
  $acc \Leftarrow 0$
  for i in $\{1 \dots 200\}$ **do**
    $\overline{\mu} \Leftarrow$ extractSample(p, μ)
    if BQ.consistency($\mathcal{N}$,$\overline{\mu}$) is **false then**
      $(pred, inc) \Leftarrow$ Alg2($\mathcal{N}$,$\overline{\mu}$)
    else
      $pred \Leftarrow$ BQ.predictions($\mathcal{N}$,$\overline{\mu}$)
    end if
    $acc \Leftarrow acc+$ agreement$(pred, \mu \setminus \overline{\mu})$
  end for
  $avgAcc \Leftarrow acc/200$
  return *avgAcc*

Alg. 7 uses the Bioquali package to validate the predictions. We applied this program to the *E. coli* influence graph and the datasets $A2$ and B. The results are shown in Table 5.6. The same steps were applied to the ASP program, but the instruction `Alg2` was replaced by the ASP program with its different repair operations. The results of the ASP validation are also shown in Table 5.6. Recall that the ASP method proposed different types of repairs that could be applied either on the network or dataset: 'e' stands for flipping edge signs, 'i' stands for making vertexes as inputs, and 'v' stands for flipping observation signs.

Regarding prediction accuracies, shown in Table 5.6, they are consistently higher than 90 percent for ASP, meaning that predicted variations and experimental observations correspond in most of the cases. Predictions accuracies for Bioquali exhibit lower

Table 5.6: Prediction accuracy on Exponential-Stationary growth shift and Heatshock data using two methods for automatic correction of inconsistencies. One based on TDDs, implemented using Bioquali. The second based on ASP.

		Prediction accuracy									
		Exponential vs Stationary					Heatshock				
	Repair	3%	6%	9%	12%	15%	3%	6%	9%	12%	15%
ASP	e	90.93	91.98	92.42	92.70	92.81	91.87	92.93	92.92	92.83	92.71
	i	90.93	91.98	92.42	92.70	92.81	91.93	92.90	92.94	92.87	92.76
	v	90.99	92.05	92.44	92.73	92.89	92.29	93.27	93.88	94.27	94.36
	e i	91.09	91.90	92.57	93.03	93.19	91.99	92.49	91.16	93.62	94.44
	e v	90.99	92.03	92.50	92.82	92.94	92.30	93.37	93.66	94.36	94.35
	i v	90.99	92.03	92.42	92.71	92.87	92.24	93.34	93.90	94.26	94.38
	e i v	91.35	92.29	92.52	93.04	—	92.26	93.04	91.78	—	—
Bioquali	—	65	70	72	73	73	76	80	78	77	76

percentage (on average 74%), this is expected since Bioquali removes all the inconsistent edges without a global optimization. ASP accuracies increase with sample size, while the choice of admissible repair operations does not exhibit much impact. Despite of this, we still observe that individual operation 'v' yields higher accuracy than 'e' and 'i'. Interestingly, this gap is greatest for the larger samples of Heatshock, where 'v' also has a lower prediction rate.

The observation that repair operation 'v' yields better prediction accuracy than 'e' (flipping edge labels) or 'i' (making vertexes input), particularly with Heatshock, suggests that repairing the data is more appropriate than repairing the model wrt the datasets we consider. In fact, operations 'e' and 'i' here correspond to network corrections, aiming at heterogeneous or missing regulations, respectively.

Biological interpretation of the results 5.7. *The high percentage of accuracy (90%) obtained from the prediction after correcting the minimal number of inconsistencies using ASP, confirms the relevance of the 11 minimal corrections (*cf. *Table 5.5) needed to conciliate the E. coli influence graph with the exponential-stationary microarray dataset.*

Biological interpretation of the results 5.8. *It is known that edge labels (activation or inhibition) are well-curated in RegulonDB, while the completeness of the network was questioned [EBK+08]. Nonetheless, the prediction accuracies we obtained indicate that the E. coli network model may not be perfect but still more reliable than experimental data, which is prone to be noisy.*

5.3.4 Discussion

We showed in the last sections how to deal with multiple inconsistencies, which are generally found when confronting large-scale regulatory data wrt genome-wide datasets. In Chapter 3 we proposed two methods for automatic correcting the network/dataset inconsistencies. In the last sections we showed and validated their results on a real biological example.

In Section 3.2.3 we introduced repair-based reasoning techniques for computing minimal modifications of biological networks and experimental profiles to make them mutually consistent. As a final application, we provided an approach to predict unobserved data even in case of inconsistency. In the last sections, we evaluated this approach on real biological examples and showed that predictions on the basis of cardinality-minimal repairs were highly accurate. This is of practical relevance because genetic profiles from DNA microarrays tend to be noisy and available biological networks far from being complete. It proposes an automatic framework that can be used as a start point to perform biological manipulations or research in order to complete the regulatory information available in databases such as RegulonDB.

5.4 Inferring the roles of TFs in the unsigned *E. coli* network

In this section we show the results obtained after applying the TF role inference approach to the unsigned *E. coli* influence graph. On that account, we removed the edge signs from the *E. coli* influence graph, and we collected two types of datasets: *(i)* computationally generated and *(ii)* multiple real datasets of *E. coli* genes expression. Afterwards, we applied to the unsigned *E. coli* topology and to the multiple collected datasets the TF role inference program in order to predict the missing signs of the edges. The TF role inference program was based on TDDs (*cf.* Alg. 3) and ASP. Our predictions on the signs of the network edges were compared to the edge signs reported in RegulonDB. We published part of this work in [VGLB+08].

5.4.1 Multiple datasets used in TF role inference analysis

For predicting unknown TF roles in an influence graph we require multiple datasets. Thus, we generated/collected the following datasets of *E. coli* gene variations:

(A) *Complete expression profiles.* This compendium was generated computationally. We used the signed *E. coli* influence graph extracted from RegulonDB in 2006, composed of TF- and σ-gene regulations (1529 nodes, 3802 edges). We simulated the effect of all possible consistent perturbations. More precisely, a perturbation experiment is represented by a set of gene expression variations $\{X_i\}$, where i represents the i^{th} network product and goes from $\{1, \ldots, 1529\}$. This set of observations is not entirely random, for its observations are constrained by the consistency equations of the TF role inference problem (*cf.* Equation 2.13).

(B) *Real compendium.* We collected a compendium of expression profiles publicly available in [FHT+07, CHG+06]. Several datasets were available, including a reference condition. When datasets were time series, we considered that each time series ends with steady state and we used the last state in the time series. Then, we sorted the measured genes in four classes: 2-fold up-regulated, 2-fold down-regulated, non-observed, and zero variation. We considered only the two first

classes and mapped them into $\{+, -\}$ changes. For some network edges, neither the input nor the output may be observed in some experiments. Altogether, we gathered 61 different experiments corresponding to over-expression, gene-deletion, and stress perturbation conditions. We verified, for all the experiments, that they correspond to the comparison between one perturbed condition against a control condition with identical levels in all chemical components except for the one altered in the perturbed condition.

5.4.2 Stress perturbation experiments: how many do you need?

For any given network topology, even when considering all possible experimental profiles, there are edge signs that cannot be determined. The TF role inference has thus a theoretical limit, referred here as *theoretical percentage of recovered edge signs*, that is unique for a given network topology. If only some perturbation experiments are available, and/or data is missing, the percentage of inferred edge signs will be lower. For a given number N of available expression profiles, *the average percentage of recovered edge signs* is defined over all sets of N different expression profiles consistent with the qualitative constraints of the unsigned influence graph (*cf.* Equation 2.13).

In this section we present the results obtained for the calculation of the theoretical and the average percentages of recovered signs for the unsigned *E. coli* influence graph, extracted from RegulonDB in 2006 consisting of TF- and σ-gene regulations (1529 nodes, 3802 edges).

The *theoretical maximum percentage of inference* is given by the number of signs that can be recovered using the *complete expression profiles* (*cf.* compendium A in Section 5.4.1). We computed this maximum percentage using first TDDs (*cf.* Alg. 3) to find most (if not all) of the predicted sings of edges. Then, ASP was used to find exactly all the remaining signs of edges. We found that *at most* 40.8% of the signs in the network edges can be predicted, corresponding to 1551 edges (M_{max}).

However, this maximum can be obtained only if all conceivable (more than 2^{50}) perturbation experiments are done, which is in practice not possible. We performed computations to understand the influence of the number of experiments (N) on the inference. For each value of N (from 5 to 200), we generated 100 sets of N *complete* random expression profiles and performed our algorithm for each set. Then, the percentage of inference was calculated as a function of N. The resulting statistics are shown in Fig. 5.15.

We can obtain a theoretical formula explaining the saturation aspect of the curve in Fig. 5.15. Let us suppose that the network contains M_1 single incoming regulations. These can be inferred with probability one from only one experiment. Let us suppose a second category of interactions, which signs are inferred with probability p $(0 < p < 1)$ on average, per experiment. This implies that the average number of inferred signs for one experiment is $M(1) = M_1 + pM_2$, where M_2 is the number of interactions in the second category. Supposing now that inference failures are independent for different experiments, we obtain the average number of inferred signs for N experiments:

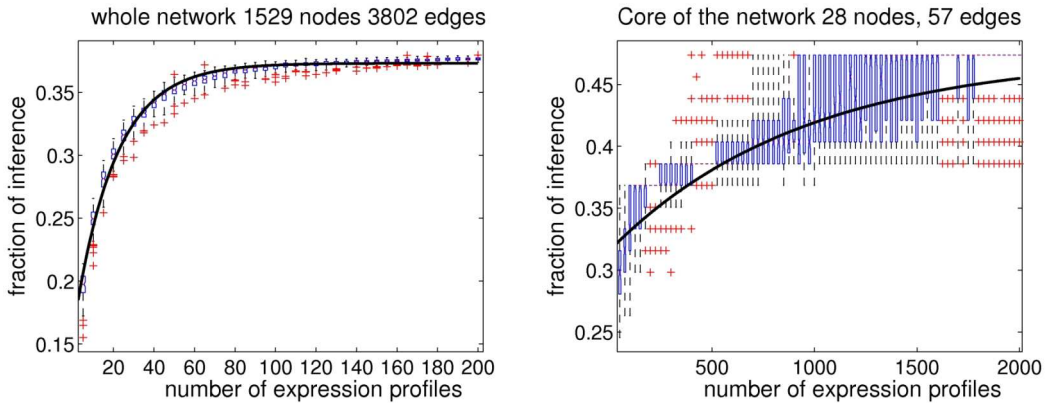


Figure 5.15: (Both) Statistics of the TF role inference process on the unsigned *E. coli* influence graph using N *complete expression profiles* (*cf.* compendium A). The Y-axis refers to the percentage of recovered edge signs. The continuous line corresponds to the theoretical formula $Y = M_1 + M_2(1-(1-p)^X)$; M_1 denotes the number of single incoming regulations inferred with probability one from any complete profile, and M_2 denotes the number of signs inferred with a probability p $(0 < p < 1)$ per experiment.
(Left) Statistics using the *whole E. coli* regulatory network. We estimated that at most 37.3% of the network edges can be inferred from a limited number of different complete profiles. Among the inferred regulations, we estimated to $M_1 = 609$ the number of signs inferred with probability one from any complete expression profile. The remaining $M_2 = 811$ signs are inferred with a probability which average is $p = 0.049$ per experiment. Hence, 30 perturbation experiments are enough to infer 33% of the network. (Right) Statistics using only the *core* of the former graph. We estimated $M_1 = 18$ and $M_2 = 9$, implying that the maximum rate of inference is $47,4\%$. Since $p = 0.0011$, the number of expression profiles required to obtain a given percentage of inference is greater than in the case using the whole network ($N = 100$ to infer 33% of the network).

$M(N) = M_1 + M_2(1-(1-p)^N)$. In general, we have $M_1 + M_2 < E$ (E is the total number of edges), meaning that there are edges which signs cannot be inferred.

For the whole *E. coli* network it appears that a few expression profiles are enough to infer a significant percentage of the network. More precisely, 30 different expression profiles may be enough to infer one third of the network (1267 regulatory roles). Adding more expression profiles continuously increases the percentage of inferred signs. For $N > 100$ we are practically on the plateau close to 37.3% (this corresponds to $M = 1420$ signed regulations).

According to our estimates the position of the plateau is $M = M_1 + M_2 = 1420$ which is smaller than the theoretical maximum $M < M_{max}$. The difference, although negligible in practice (to obtain M_{max} one has to perform $N > 2^{50}$ experiments), suggests that the plateau has a very weak slope. This means that contributions of different experiments to sign inference are weakly dependent. The values of M_1, M_2, and p estimate the efficiency of our method: large p, M_1, and M_2 mean small number of expression profiles needed for inference.

Biological interpretation of the results 5.9. *To recover one third of the edge signs of the* E. coli *influence graph, composed of TF- and σ-gene interactions, it is needed 30 different expression profiles. These profiles must correspond to consistent observations*

of all the network products.

5.4.3 Inferring the TF roles of the network core

The core of the *E. coli* influence graph used in our previous calculation had 28 nodes and 57 edges (*cf.* Section 5.1.3). In the previous section we showed the TF role inference results for this smaller graph. Not surprisingly, we noticed a rather different behavior when inferring signs on a core graph than on a whole graph as demonstrated in Fig. 5.15. In the former case, we needed much more experiments for the inference since the sets of expression profiles contained from $N = 50$ to 2000 random profiles.

Two observations may be concluded. First, a greater number of experiments is required to reach a comparable percentage of inference; the value of p is smaller than for the whole network. This confirms that the core is more difficult to infer than the rest of the network. Second, Fig. 5.15 displays a much less continuous behavior for the core. More precisely, when using the core, different perturbation experiments have strongly variable impact on sign inference. For instance, the experimental maximum percentage of inference (27 signs over 57) can be obtained already from about 400 expression profiles, yet, most of the datasets with 400 profiles infer only 22 signs.

This suggests that not only the core of the network is more difficult to infer, but also that a brute force approach (multiplying the number of experiments) may fail as well. This situation encourage us to apply experiment design and planning, that is, computational methods to minimize the number of perturbation experiments while inferring a maximal number of regulatory roles.

This also illustrates why our approach is complementary to dynamical modeling. In the case of large scale networks, when an interaction stands outside the core of the graph, an inference approach is suitable to infer the sign of the interaction. However, when an interaction belongs to the core of the network, more complex behaviors occur (*e.g.* influences that depend on activation thresholds), thus, a precise modeling of the dynamical behavior of this part of the network should be performed [DJ02].

Biological interpretation of the results 5.10. *The number of inferred edge signs depends on the topology of the network. For a more connected topology, such as the core of the E. coli network, more experiments are needed to infer the sign of its edges.*

5.4.4 Influence of missing data

In the previous sections we assumed that all products in the network were observed. That is, in each experiment each node is assigned a value in $\{+, -\}$. However, in real measurement devices, such as expression profiles, a part of the values is discarded due to technical reasons. A practical method for network inference should cope with missing data.

We studied the impact of missing values on the percentage of inference. For this, we considered a fixed number of expression profiles ($N = 30$ for the whole *E. coli* network, $N = 30$ and $N = 200$ for its core). Then, we randomly discarded a growing

percentage of observed products in the profiles, and computed the percentage of inferred regulations. The resulting statistics are shown in Fig. 5.16.

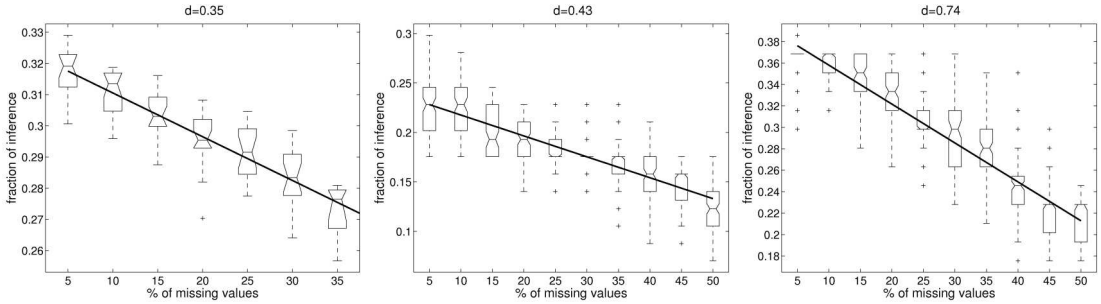


Figure 5.16: (All) Statistics of the TF role inference process on the *E. coli* influence graph (1529 nodes, 3802 edges) from *partial* expression profiles. In these experiments, the number of experiments N is fixed. The continuous line corresponds to the theoretical prediction $M_i = M_i^{max} - d * f * M_{total}$, where M_i^{max} is the number of inferred edge signs from *complete* expression profiles, d is the number of interaction signs no longer inferred when a node is not observed, f is the fraction of unobserved nodes, and M_{total} is the total number of nodes.
(Left) Statistics for the whole network. We used 30 sets of artificial expression profiles ($N = 30$). We estimated $d = 0.14$, meaning that on average we lose one interaction sign for about 7 missing values in the profiles. (Middle) Statistics for the core network ($N = 30$). We estimated $d = 0.21$. The core of the network, however, is more sensitive to missing data. (Right) Statistics for the core network ($N = 200$). We estimated $d = 0.36$. Hence, increasing the number of expression profiles increases the sensitivity to missing data.

In both cases (whole network and core) the dependency between the average percentage of inference and the percentage of missing values is qualitatively linear. Simple arguments allow us to find an analytic dependency. If not observing one node of the network implies losing information on d interaction signs, we are able to obtain the following linear dependency $M_i = M_i^{max} - d * f * M_{total}$, where M_i^{max} is the number of inferred interactions for complete expression profiles (no missing values), f is the fraction of unobserved nodes, and M_{total} is the total number of nodes. In order to keep M_i non negative, d must decrease with f. Our numerical results imply that the constancy of d and the linearity of the above dependency extend to rather large values of f. This indicates that our qualitative inference method is robust enough for practical use. For the whole network we estimated $d = 0.14$, meaning that on average we lose one interaction sign for about 7 missing values. However, for the same number of expression profiles, the core of the network is more sensitive to missing data (the value of d is larger, it corresponds to lose one sign for about 4.8 missing values). For the core, increasing the number of expression profiles increases d and hence the sensitivity to missing data.

Biological interpretation of the results 5.11. *When studying the influence of missing observations on the datasets, we concluded that for the E. coli influence graph we miss one edge sign prediction for about 7 missing observed values on average.*

5.4.5 TF role inference with a real compendium of expression profiles

For this analysis we used the *E. coli* influence graph obtained from RegulonDB in 2007 (1415 nodes, 2899 edges). This graph was composed only of TF-gene regulations. We used a real compendium of *E. coli* gene expression profiles composed of 61 datasets (*cf.* compendium B in Section 5.4.1).

We applied the TF role inference program based on TDDs (*cf.* Alg 3) to the unsigned *E. coli* influence graph with the 61 datasets. In these datasets, 12.9% of the network products were observed on average. When summing all the observations, 17.2% (497) of the network edges (input and output) were observed in at least one expression profile. We predicted 152 edge signs in the network (30% of the edges observed at least once). We compared the predictions to the known interaction signs: 28.3% of the predictions were false predictions. Sources of errors may lie on non-modeled interactions (possibly effects of absent sigma-factors), or in using experiments on different *E. coli* strains.

We filtered our predictions according to their reliability. On that account, we used a filtering parameter, which is a positive integer k representing the number of different experiments with which the predicted edge sign is consistent. For a filtering value k, all the predictions that are consistent with less than k profiles are rejected. We used values for the filtering parameter $k = \{1, \cdots, 5\}$. Filtering improves our prediction quality allowing us to retain only reliable predictions. Thus, for $k = 5$, we predicted 41 edge signs, of them, only 1 was an incorrect prediction (2.5% of false prediction). We conclude that filtering is a good way to strengthen our predictions even when the model is not precise enough. We illustrated the effect of the filtering process in Fig. 5.17A.

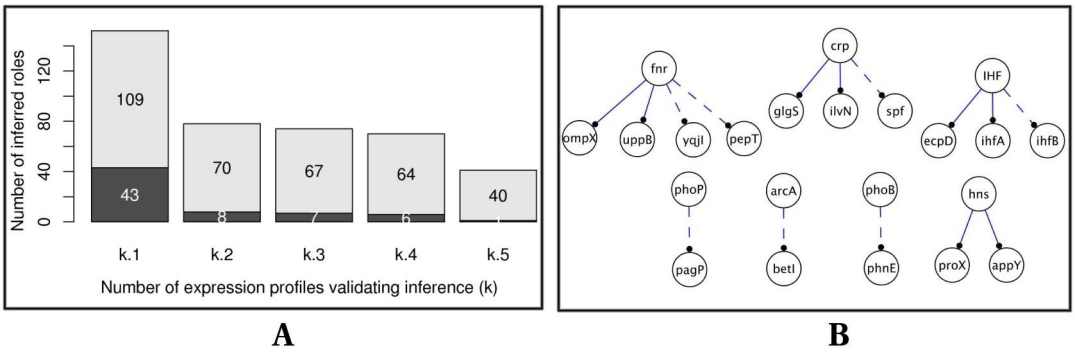


Figure 5.17: **A.** Results of the inference algorithm applied to *E. coli* network with a compendium of 61 experiments. The dark and light regions of the bars correspond to false positive and validated predictions, respectively. Without filtering, there are 28.3% of false positives. With filtering – keeping only the sign predictions confirmed by k different experiments – the rate of false positives decreases to 2.5%. **B.** Ambiguous *E. coli* interactions with the 61 datasets. For each interaction there exist at least two datasets that do not predict the same sign on the interaction.

Our algorithm also detected seven inconsistent graphs in the network (see Fig. 5.17B). A list of experimental assays that yield inconsistencies on each interaction

is given in the Supplementary Web site of [VGLB+08] at: `http://www.irisa.fr/symbiose/interactionNetworks/supplementaryInference.html`. This analysis shows that there exist non-modeled interactions that balance the effects on the targets in the detected inconsistent graphs.

Biological interpretation of the results 5.12. *We inferred 30% of the E. coli edge signs which input and output are simultaneously observed in at least one experiment, using 61 real expression profiles. 71% of these edge predictions were in agreement with the edge signs reported in RegulonDB.*

5.4.6 Discussion

In principle, inferring the functional effect of regulations could be done using general reconstruction methods. The most outstanding approaches in this domain include Bayesian networks [FLNP00], linear ordinary differential equations (ODE) [DBTG+05, BDGdB06] and correlation/causal networks [NTIM05, XvdL05, MNB+06] (see [BBAIdB07] for a review and a comparison on several datasets). These are quantitative methods which are carefully designed to cope with the high level of noise that is generally observed in expression data. They rely either on an explicit parametric modeling of noise distribution (like in Bayesian networks), or on robust statistical estimators for the network and its kinetic parameters. The main limitation of these approaches is the number of independent samples they require in order to be properly used. It is often stated [BBAIdB07, MNB+06] that a minimum of 100 to 300 expression profiles are needed for the estimation procedure. While there exists a couple of datasets of such size, the usual number of available profiles for a given biological system is much smaller. Our approach is meant to be used when the number of profiles ranges from 1 to a couple of hundreds, and should thus be seen as complementary to quantitative methods. Indeed our simulations on the *E. coli* network showed that one can characterize about 30% of the regulations from 30 expression profiles. We additionally showed that this is close to the theoretical limit of our approach. This result was confirmed using real expression data on the same network: we inferred 30% of the regulations which input and output are simultaneously observed in at least one experiment, using 61 expression profiles.

Using simulations, we evaluated the dependence between the number of available expression profiles and the number of signs that can be inferred from them. Not surprisingly, we noticed that the topology of the regulatory network has a strong influence on the estimated relationship. This was illustrated by computing statistics on both a complete regulatory network and its core. The complete network is characterized by an over-representation of feedback-free regulatory cascades, which are controlled by a small number of TFs. In this setting, the number of inferred signs grows almost continuously with the number of observations. In contrast, the core network does not obey the simple law "the more you observe, the better", some expression profiles being clearly more informative than others. Additionally, in these core networks an unfeasible number of experiments is necessary to infer a small number of signs with high probability. For these core networks, two different strategies may be adopted. First, to build a more

accurate model for these restricted subnetworks using dynamic modeling techniques [DJ02]. Second, to develop experiment planning in our qualitative framework.

Chapter 6

Application to Eukaryote networks

In the previous chapter we applied our modeling approach to the *E. coli* transcriptional network. Although this organism is very well studied, its transcriptional regulatory machine is simplified with respect to eukaryotic organisms. One major difference between prokaryotes and eukaryotes is the existence of the nucleus membrane, which divides the place where transcription and translation occurs in eukaryotic cells. Another difference is that in eukaryotic cells much more post-transcriptional processes take place after the mRNA production. In prokaryotic cells mRNA usually stays unchanged.

In this chapter we will discuss the results obtained when dealing with two eukaryotic regulatory network models. The first one is the transcriptional regulatory network of *S. cerevisiae*. The second one is the signaling network of the EWS-FLI1 human oncogene. We present the different analyses proposed over these more complex systems, as well as the biological impact and validation of our results. In our first study we show a comparison between the TF role inference approach when applied to the *S. cerevisiae* eukaryotic regulatory model with respect to the *E. coli* prokaryotic model; part of the results of this work were published in [VGLB+08]. In our second study, we describe how we adapted our approach in order to consider specific representations of post-translational phenomena.

6.1 *S. cerevisiae* transcriptional network

The budding yeast *S. cerevisiae* is one of the most intensively studied eukaryotic model organisms in molecular and cell biology. On that account we used its transcriptional regulatory network for validating our TF role inference approach. In this section we present the obtained results; we published them in [VGLB+08].

6.1.1 Constructing the *S. cerevisiae* unsigned influence graphs

In the following, we will briefly review the available sources that were used to build the unsigned *S. cerevisiae* influence graph. The experimental dataset proposed in [LRR+02] is widely used in the network reconstruction literature. It is a study conducted under nutrient rich conditions, and it consists of an extensive ChIP-chip screening of 106

TFs. Estimations regarding the number of yeast TFs that are likely to regulate specific groups of genes by direct binding to the DNA vary from 141 to 209, depending on the selection criteria. In follow-up papers of this work, the ChIP-chip analysis was extended to 203 yeast TFs in rich media conditions and 84 of these regulators in at least one environmental perturbation [HGL+04]. Analysis methods were refined in 2005 by MacIsaac and colleagues [MWG+06]. Other studies continued to work on this network using different approaches [XvdL05, SSR+03, NTIM05, BBAIdB07]. Here we selected two of these sources ([LRR+02] and [MWG+06]) to build four influence graphs for *S. cerevisiae*. The complete list of interactions of these networks is provided in the Supplementary Web site of [VGLB+08] available at: `http://www.irisa.fr/symbiose/interactionNetworks/supplementaryInference.html`. The four influence graphs built for our analysis were:

(A) The first influence graph consists of the core of the transcriptional ChIP-chip regulatory network produced in [LRR+02]. Starting from the full network with a P-value of 0.005, we reduced it to the set of nodes that have at least one output edge. This network was already studied in [KPST03]. It contains 31 nodes and 52 interactions.

(B) The second influence graph contains all the transcriptional interactions between TFs shown by [LRR+02] with a P-value below 0.001. It contains 70 nodes and 96 interactions.

(C) The third influence graph is the set of interactions among TFs as inferred in [MWG+06] from sequence comparisons. We considered the network corresponding to a P-value of 0.001 and 2 bindings. It contains 83 nodes and 131 interactions.

(D) The last influence graph contains all the transcriptional interactions among genes and regulators shown by [LRR+02] with a P-value below 0.001. It contains 2419 nodes and 4344 interactions.

6.1.2 Multiple datasets used in the TF role inference process

The following sets of gene expression profiles were used in the *S. cerevisiae* TF role inference process:

(P1) *Gene-deletion profiles.* We collected 210 gene-deletion experiments available in [HMJ+00].

(P2) *Stress perturbation profiles.* This data corresponds to curated information available in the Saccharomyces Genome Database (SGD, [HBC+01]). When time series profiles were available, we selected the last time expression array. Therefore, we collected and treated 15 experiments described in Table 6.1. For each expression array we sorted the measured genes in four classes: 2-fold up-regulated, 2-fold down-regulated, non-observed, and null-variation.

Table 6.1: List of genome expression experiments on *S. cerevisiae* used in the sign inference process. All experiments contain information on steady state shift and their curated data is available in the Saccharomyces Genome Database.

Experiment ID	Description	Reference
E1	Diauxic Shift	[DIB97]
E2	Sporulation	[CDE+98]
E3	Expression analysis of Snf2 mutant	[SIBW00]
E4	Expression analysis of Swi1 mutant	[SIBW00]
E5	Pho metabolism	[ODB00]
E6	Nitrogen Depletion	[GSK+00]
E7	Stationary Phase	[GSK+00]
E8	Heat Shock from 21°C to 37°C	[GSK+00]
E9	Heat Shock from 17°C to 37°C	[GSK+00]
E10	Wild type response to DNA-damaging agents	[GHM+01]
E11	Mec1 mutant response to DNA-damaging agents	[GHM+01]
E12	Glycosylation defects on gene expression	[CSG+04]
E13	Cells grown to early log-phase in YPE	[RH06]
E14	Cells grown to early log-phase in YPG	[RH06]
E15	Titratable promoter alleles - Ero1 mutant	[MDH+04]

6.1.3 Inference process with gene-deletion expression profiles

We applied the TF role inference program based on TDDs (*cf.* Alg. 3) to the *S. cerevisiae* influence graph D, composed of 2419 nodes and 4344 edges. We used the panel of expression profiles P1 (210 experiments). The information given by this panel is quite small, since 1.6% of all the products in the network were observed on average, and 12% of the edges (input and output) of the network were observed in at least one expression profile. Using this data we inferred 162 edge signs. We validated our prediction with a literature-curated network on Yeast [NGBBK02]. We found that among the 162 sign-predictions, 12 were referenced with a known interaction in the database, and 9 with a good sign.

Gene-deletion expression profiles were used in order to compare our results with the path analysis methods [YIJ04, YMM+05], since the latter can only be applied to knock-out data. Other sign-regulation inference methods needed either other sources of gene-regulatory information (promoter binding information, protein-protein information), or time-series data to be performed [SSR+03, SE05, BBAIdB07].

We compared the inference results for both methods, our approach and the path analysis method, obtaining in the latter that 234 roles of widely connected paths were inferred, whereas with our method 162 roles were inferred, mainly localized in the branches of the network. Both results intersected on 17 interactions and no contradiction in the inferred role was reported. This suggests that our approach is complementary to the path analysis methods. Our explanation is as follows: in [YIJ04, YMM+05] network inference algorithms identify probable paths of physical interactions connecting a gene knock-out to genes that are differentially expressed as a result of that knock-out. This leads to a search for the smallest number of interactions that carry the largest information in the network. Hence, inferred interactions are located near the core of

the network, but not exactly in the core. On the contrary, as we already mentioned, the combinatorics of interactions in the core of the network are too intricate to be determined from a few hundreds of expression profiles with our algorithm, thus, we concentrate on interactions around the core.

Biological interpretation of the results 6.1. *After applying the TF role inference approach to the S. cerevisiae network containing all the transcriptional interactions (2419 nodes and 4344 interactions) using 210 experiments, we inferred 162 edge predictions. Path analysis methods inferred on the same data 234 TF roles. Our predictions are complementary with this study, since they are located in different network regions.*

6.1.4 Inference with stress perturbation expression profiles

In order to overcome the problem exposed using the small amount of information of the P1 profiles, we used stress perturbation experiments. This data corresponds to the panel of expression profiles P2 (15 experiments). We executed our TF role inference program on the four *S. cerevisiae* influence graphs described in Section 6.1.1. We identified inconsistent graphs, as well as predicted edge signs. For the influence graph D, we filtered the edge predictions using $k = 3$. In Table 6.2 we illustrate our results. The total inference rate was obtained by adding the number of predicted edge signs to the number of non-repeated interactions in the inconsistent graphs detected, and dividing it by the total number of edges in the network. Depending on the network, the inference rate varies from 19% to 37%. Thus, they are similar to the theoretical rates obtained for the *E. coli* network (*cf.* Section 5.4.2) even with a small number of perturbation experiments.

Table 6.2: TF role inference process applied to four *S. cerevisiae* influence graphs. 2-fold significant observations of 15 experiments were used for the inference. The *In/Out observed simultaneously* rate refers to the sign inference rate if all observations of the in/out nodes of one edge lead to predictions. The *Inferred signs* are the number of signs fixed in an unique $\{+,-\}$ way by all the experiments.

Influence graph	Nodes	Edges	Average observed nodes	In/Out observed simultan.	Inferred $\{+,-\}$ edge signs	Number of inconsistent edges	Total Inference
(A)	31	52	28%	88%	11	3	26.8%
(B)	70	96	26%	72%	29	7	37.4%
(C)	83	131	33%	69%	21	4	19%
(D)	2419	4344	30%	52%	no filter: 631 filter k = 3: 198	682	32%

We validated the inferred interactions, for the case of the *S. cerevisiae* influence graph D, by comparing them with the literature-curated network published in [NGBBK02]. Of the 631 $\{+,-\}$ signs of edges predicted when no filtering was applied, 23 were referenced with a known interaction in the database, and 16 with a good sign. Furthermore, of the 198 interactions predicted with a filter parameter $k = 3$, 19 were referenced with a known interaction in the database, and 18 with a good sign. As in the case of *E. coli*, we conclude that filtering is a good way to strengthen our predictions. In Fig. 6.1 we illustrate the inferred interactions for Network B.

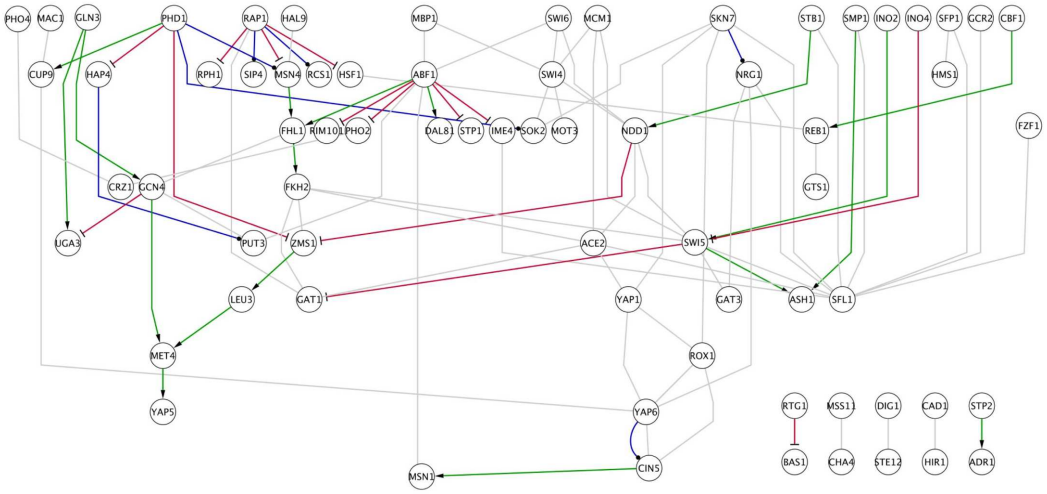


Figure 6.1: *S. cerevisiae* influence graph B. Only interactions among TFs were considered. A total of 29 interactions were inferred. Green and red arrows correspond to inferred activations and repressions, respectively. Blue arrows correspond to the inconsistencies detected.

As already mentioned, the TF role inference algorithm identified a large number of ambiguities. The inconsistent edges found for the influence graphs A, B, and C are shown in Table 6.3. These inconsistencies refer to edges which sign is set to '+' by one experiment and '−' by another. The exhaustive list of the inconsistencies for the influence graph D is given in the Supplementary Website of [VGLB+08]. For each inconsistency a precise biological study of the species should allow to understand the origin of the ambiguity: erroneous expression data, missing interactions in the model, or context-dependent regulations.

Biological interpretation of the results 6.2. *The TF role inference results, obtained when using a smaller number of expression profiles (15), shows a higher rate of predicted edge signs in the S. cerevisiae influence graph. This is because the small set of expression profiles was more complete. The inference rate varied from 19% to 37%. We obtained a 69% of accuracy for the larger influence graph of S. cerevisiae. However, the reference network we compared with contained very few interactions. This result is biologically relevant because for S. cerevisiae it does not exist so well curated databases as RegulonDB, where the signs of the interactions are available.*

Biological interpretation of the results 6.3. *The number of inconsistencies found after applying the TF role inference process to the S. cerevisiae influence graph was higher than in the case of E. coli. Even if in both approaches 2-fold significant observations were used. We justify this difference because in eukaryotic organisms more post-transcriptional interactions are present. Hence, conciliating mRNA measurements for S. cerevisiae is a more difficult task than for E. coli if we only consider a network with transcriptional interactions.*

Table 6.3: Inconsistent interactions found for three influence graphs of *S. cerevisiae*. For each inconsistent edge we list two experiments that infer a different role of regulation for it.

Influence graph	Actor	Target	Experiment 1	Experiment 2
(A)	YAP6	CIN5	Sporulation [CDE+98]	Stationary Phase [GSK+00]
	GRF10	MBP1	Stationary Phase [GSK+00]	Mec1 mutant [GHM+01]
	PDH1	MSN4	Nitrogen Depletion [GSK+00]	Heat shock 21°C to 37°C [GSK+00]
(B)	YAP6	CIN5	Sporulation [CDE+98]	Stationary Phase [GSK+00]
	RAP1	SIP4	Sporulation [CDE+98]	Diauxic shift [DIB97]
	SKN7	NRG1	Stationary Phase [GSK+00]	Diauxic shift [DIB97]
	PHD1	SOK2	Heat shock 21°C to 37°C [GSK+00]	Stationary Phase [GSK+00]
	RAP1	RCSK1	Wild type + Heat [GHM+01]	Respiratory growth [RH06]
	PHD1	MSN4	Nitrogen Depletion [GSK+00]	Heat shock 21°C to 37°C [GSK+00]
	HAP4	PUT3	Diauxic shift [DIB97]	Snf2 mutant, YPD [SIBW00]
(C)	SWI5	ASH1	PHO pathway [ODB00]	Stationary Phase [GSK+00]
	SKN7	NRG1	Stationary Phase [GSK+00]	Nitrogen Depletion [GSK+00]
	NRG1	YAP7	PHO pathway [ODB00]	Respiratory growth [RH06]
	NRG1	GAT3	Glycosylation [CSG+04]	Respiratory growth [RH06]

6.1.5 Discussion

The problem of inferring functional effects of transcription factors was specifically addressed by Yeang and colleagues [YIJ04, YMM+05], using a probabilistic discrete model. In this approach, one identifies probable paths of physical interactions connecting a gene knock-out to genes that are differentially expressed as a result of that knock-out. Predictions correspond to the signs found in models of maximum likelihood. More generally, most reconstruction methods are based on computing an "optimal" model with respect to the data. This raises two main issues. First, the underlying optimization problems are often non-convex, and finding a global optimum is a very difficult computational task. In practice, most algorithms only guarantee to find a local optimum, which should be cautiously examined before being reported as a prediction. Second, even if a global optimum is found, it is important (but computationally difficult) to check that there is no slightly sub-optimal model that yields very different predictions. In other terms, it is necessary to evaluate the *robustness* of the predictions. In our approach, we describe the (possibly huge) set of models that are consistent with the data, then look for invariants in this set. This means that our predictions are compatible with *all* feasible models. In order to cope with experimental noise, we combine this strategy with a filtering procedure, which selects predictions that agree with a minimal number of expression profiles. This led us to very accurate predictions, as it was shown on yeast data. We compared our inference approach to the path analysis method by Yeang and colleagues [YIJ04, YMM+05] and found that both algorithms infer a similar number of regulations, and that the predictions coincide. We noticed that the predictions are located in different parts of the network, depending on the algorithm: path analysis tends to infer signs in highly connected regions, while our approach infer signs on regulations acting on small in-degree nodes. Another difference is that path analysis requires expression

profiles from gene-deletion experiments, whereas our method gives better results with stress perturbation experiments (though it can be applied to both types of experiments).

6.2 EWS-FLI1 signaling network

In this section we present the results obtained after applying the consistency-check process to the influence graph of the EWS-FLI1 signaling network. This network describes the effects of a fusion oncogene EWS-FLI1 on cell cycle, inducing young adult cancers [DZP+92]. Time series data and annotated gene regulatory network were produced at the Institute Curie[1] [SZN+08]. We submitted this work to the IEEE/ACM "Transactions on Computational Biology and Bioinformatics" (IEEE-TCCB) journal.

6.2.1 Constructing the EWS-FLI1 influence graphs

The signaling network of EWS-FLI1 is a very well annotated regulatory model, which contains precise information about the types of network interactions. Using this information we built two influence graphs. In the first one, called *generic*, the influences arriving to a node were modeled using the generic qualitative constraint: $t \simeq GEN_T$ (*cf.* Equation 2.2). In the second one, called *refined*, we precised the qualitative function F_T for certain nodes in the network. In addition, in the refined influence graph we introduced a precise modeling of the phosphorylated molecules.

Generic influence graph – native

In an independent work, an annotated gene regulatory and signaling model involving 130 genes, including EWS-FLI1, was designed from the genes that responded to EWS-FLI1 as follows. Using information from TRANSPATH and literature, interactions were selected to describe signal pathways that regulate key functions involved in tumor progression (cell cycle phase transitions, apoptosis, and cell migration) [SZN+08].

From this native model we designed an influence graph. In order to capture the effect of post-translational regulations, each node of the native model was divided into two molecular species: *mRNA nodes* and *active protein nodes*. Influences between nodes were set according to the nature of each interaction (transcriptional or post-translational) provided by the annotation of the native model. This annotation was also used to assign $\{+, -\}$ values to all the graph edges. In this influence graph the variation of a network product was modeled using the generic qualitative constraint $t \simeq GEN_T$. The resulting influence graph contained 287 nodes and 644 signed edges. The nodes in this graph corresponded to: mRNAs, active-proteins, active-protein-complexes, and phenotypes as 'cell cycle' and 'apoptosis'.

[1] http://bioinfo-out.curie.fr/projects/sitcon/

Refined influence graph – after adding precise qualitative functions

In order to constrain more the behavior of the system, we designed a new influence graph based on the generic influence graph provided above. This new graph had the following characteristics:

- The phosphorylated proteins were modeled according to their change in state after the phosphorylation. That is, if A is a protein that becomes active when phosphorylated by B, we added in our influence graph the interaction '$B \rightarrow A^{act}$ +'. In the opposite case, we added the interaction '$B \rightarrow A^{act}$ −'. Recall that our influence graph was composed only of mRNA and active-protein nodes.
- The variation of 36 network products was modeled using an specific (non-generic) qualitative function. In Table 6.4 we show a summary of the four qualitative functions introduced in the system of constraints. For more details regarding the modeling choice of these functions refer to Section 2.3.2.2.
- The nodes that were neither phosphorylated proteins nor receiving a specific qualitative function, were modeled using the generic constraint.

Notice that instead of the generic function GEN that is always satisfied, the new functions that we introduced are designed according to the biological literature on the products implied in the network that we are currently considering. Some rules may be quite generic and used in other contexts, but this deserves a specific mathematical study which is not our purpose presently. The refined influence graph of the EWS-FLI1 network had 296 nodes (36 received a non-generic qualitative function), and 430 $\{+, -\}$ edges.

Table 6.4: Non-generic qualitative functions added to the refined influence graph of the EWS-FLI1 signaling network.

	Qualitative function	Example	Description	Reference
(1)	Protein complex formation	$F_{CCND-CDK} = ccnd$	Cdks are constitutively expressed and present in excess to D-type cyclins	[OS02]
(2)	Strong inhibitor	$F_{cell\_cycle\_S} = e2f \wedge \neg rb1$	Sequestration	[DeG02]
(3)	Complex inactivation	$F_{CCNE\_CDK2} = \neg wee1 \wedge (ccne \oplus cdk2)$	Proteins may hamper complex formation	[OS02]
(4)	Complex inactivation-reactivation	$F_{CCNA\_CDK2} = (ccna \oplus cdk2) \wedge (\neg cdkn1a \vee ccnd\_cdk4\_a)$	A protein-complex may be under an inactivation influence that is itself inhibited	[PRKG+99]

6.2.2 Datasets used in the analysis

We describe in the following the experimental steps considered to obtain the qualitative dataset of observations used in the consistency analysis. The A673 cell line derived

from an Ewing tumor was modified as follows: after induction by doxocyclin, a sh-RNA targeting EWS-FLI1 is produced, inactivating EWS-FLI1. This stops cell division. The gene response to EWS-FLI1 inactivation was investigated by using an Affymetrix HG-U133 Plus 2.0 microarray during 17 days on two independent clones. At day 11, cells from one clone were washed and harvested in a solution without doxocyclin. When EWS-FLI1 is reactivated, the cell division restarts. The gene response to EWS-FLI1 reactivation was investigated by using the same microarray between day 11 and day 17.

The time series data on Ewing inducible cell lines were analyzed to select genes which show a significant response on both inhibition and reactivation of EWS-FLI1. If a gene is inhibited upon the EWS-FLI1 inhibition (Day0-Day11) and reactivated with reactivation of the oncogene, and if these responses are significant, then we consider this gene to *correlate* with EWS-FLI1 behavior. Likewise, if a gene is up-regulated at EWS-FLI1 inhibition and down-regulated with EWS-FLI1 reactivation, showing the significant variation of its response, such gene is considered as *anti-correlated* with the oncogene. With the correlated and anti-correlated measured mRNAs we built two datasets of observations:

(A) *EWS-FLI1 inhibition.* This dataset corresponded to the measured variations of the mRNAs during the EWS-FLI1 inhibition. It was built by classifying as '+' the correlated mRNAs, and as '−' the anti-correlated mRNAs. It consisted of 54 $\{+,-\}$ variations of the mRNAs present in the EWS-FLI1 signaling network.

(B) *EWS-FLI1 reactivation.* This dataset represented the qualitative change in time of the mRNAs when the EWS-FLI1 oncogene was reactivated. It was obtained by reverting the signs of dataset A.

6.2.3 Studying the impact of a precise modeling on predictions

We applied the consistency-check process described in Sections 2.2.3 and 2.3 to the generic and refined influence graphs of EWS-FLI1. In our analyses we used datasets A and B (*cf.* Section 6.2.2). Both influence graphs were found to be consistent with both datasets.

Afterwards, we computed the predictions of both graphs using only dataset A. 37 nodes were predicted using the generic influence graph, while 55 were predicted using the refined influence graph. In the first case 2 mRNA nodes, 2 protein-complexes, and 33 protein activities were predicted. In the second case 6 mRNA nodes, 13 protein-complexes, and 36 protein activities were predicted. The list of new predictions obtained using the refined influence graph is shown in Table 6.5.

All predictions obtained using the generic influence graph but one (RBL1) were also predicted by using the refined influence graph. The RBL1 protein was modeled in the generic influence graph without taking into account the influences that phosphorylate it, thus its activity was easy to predict as it received only one transcriptional influence. In the refined graph, however, the active protein RBL1 received also phosphorylation

Table 6.5: New predictions obtained using the refined influence graph. The *Prediction* probability column states how relevant is our prediction wrt random data (it will be detailed in the next section). The third column specifies which nodes were modeled using a complex qualitative function.

Predicted node			Refined Model Prediction Prob.	Complex Function
CCNA2_CDC2_act	→	–	0,187	X
CCNA1_CDC2_act	→	–	0,181	X
CCNB1_CDC2_act	→	–	0,17	X
CCND3_CDK4_act	→	–	0,144	X
CCND3_CDK6_act	→	–	0,144	X
CCNA1_CDK2_act	→	–	0,142	X
CCNA2_CDK2_act	→	–	0,139	X
E2F2	→	–	0,127	X
CCND2_CDK4_act	→	–	0,086	X
CCND2_CDK6_act	→	–	0,086	X
E2F3	→	–	0,077	X
CCNB1_CDC2	→	–	0,077	X
E2F5	→	–	0,072	X
CCND1_CDK6_act	→	–	0,069	X
CCND1_CDK4_act	→	–	0,069	X
mRNA_TP73	→	–	0,002	
mRNA_APAF1	→	–	0,002	
mRNA_CDKN2A	→	–	0,001	
mRNA_RB1	→	–	0	

influences; it was not possible to predict a definite change for RBL1 from the values of these influences.

Biological interpretation of the results 6.4. *Designing specific logical rules to describe the effects of post-translational interactions significantly increased the number of predictions (+ 44.7%). Additionally, most of the new predictions were over protein or protein-complex nodes.*

6.2.4 Estimating the specificity of consistency and prediction

In order to evaluate the specificity of a consistency diagnosis or a prediction on a given node, we introduce indicators based on the predictions obtained from random datasets. The EWS-FLI1 influence graphs (*cf.* Section 6.2.1) were composed of a set of nodes V, that could be divided into three subsets: $V = S_{mrna} \cup S_F \cup S_O$, where S_{mrna} denotes the set of mRNA nodes, which variations can be observed from the time-series analyses, S_F denotes a set of fixed phenotypes, and S_O denotes the active proteins or protein-complexes. Following this partition, a set of partial observations μ also splits into three subsets: $\mu = \mu_{mrna} \cup \mu_F \cup \mu_o$. In our case the dataset of observations μ was only obtained from mRNA measurements and phenotypical observations; μ_o was empty.

Investigating the specificity of consistency and prediction requires to generate random datasets of observations. From an initial set of observations μ, m new random datasets $\mathcal{R}and_i(\mathcal{N}, \mu) = random(S_{mrna}, \mu_{mrna}) \cup \mu_F$ are produced as follows: the *random* function outputs a dataset of observations by randomly selecting $|\mu_{mrna}|$ nodes

in S_{mrna} and assigning them $\{+,-\}$ variations preserving the sign distribution in μ_{mrna}. The total set of random samples will be:

$$\mathcal{R}and(\mathcal{N},\mu) = \{\mathcal{R}and_i(\mathcal{N},\mu) \mid i = 1 \ldots m\}$$

It is possible now to evaluate the specificity of a consistency diagnostic. Let $\mathcal{C}ons \subset Rand$ denote the subset of random datasets that are consistent with the network $\mathcal{N}$. The *Consistency P-value* of a network $\mathcal{N}$ wrt μ is given by: $p_{consistent}(\mathcal{N},\mu) = \frac{\#\mathcal{C}ons}{\#\mathcal{R}and(\mathcal{N},\mu)}$.

The P-value gives hints about how a consistency diagnostic is significant. It quantifies the way the topology of the network constrains the whole set of observations. The smaller this probability is, the more distant an observation dataset will be from random with respect to the network. Notice that the consistency P-value is highly dependent on the cardinality of the initial dataset μ: if the initial dataset contains few products (even if accurately chosen with respect to biological insights), the random function will give rise to random datasets containing products that have no biological interest and provide no constraint to the network.

This default can be overcome by investigating the specificity of each prediction instead of the full consistency diagnosis. To that purpose, by using the function BQ.predictions($\mathcal{N}$,μ) introduced Section 3.1.2.1, we define the *Prediction probability* $P_{pred}(n,s)$ of a node n in $\mathcal{N}$, predicted with a $s \in \{+,-\}$ value, as follows:

$$P_{pred}(n,s,\mathcal{N},\mu) = \frac{1}{\#\mathcal{R}and(\mathcal{N},\mu)} \sum_i f_{\mathcal{N},\mu,\mathcal{C}ons_i}(n,s),$$

where $Cons_i$ represents one consistent random dataset, *i.e.* $Cons = \bigcup Cons_i$, and $f_{\mathcal{N},\mu,\mathcal{C}ons_i}(n,s)$ evaluates if a prediction (n,s) from a real dataset μ is predicted with the same sign when $\mathcal{N}$ is confronted to $Cons_i$:

$$f_{\mathcal{N},\mu,\mathcal{C}ons_i}(n,s) = \begin{cases} 1 & \text{if } (n,s) \in BQ.predictions(\mathcal{N},\mu) \cap BQ.predictions(\mathcal{N},\mathcal{C}ons_i) \\ 0 & \text{otherwise.} \end{cases}$$

In other words, the prediction probability indicates whether a given prediction on a node can be obtained in a random way or if it is a consequence of the observation dataset. With this point of view, predictions with a small prediction probability shall be considered as the more informative of the model and have the top priority to be experimentally confirmed.

Application to the EWS-FLI1 signaling network

We applied the mathematical framework explained above on both EWS-FLI1 influence graphs considering only dataset A. This dataset had 54 observations on mRNA-nodes (denoted by μ_{mrna}). Knowledge on Ewing tumour phenotypes set up 7 phenotypical nodes (S_F) to $\{+,-\}$ changes (denoted by μ_F). For both graphs, generic and refined, a set $\mathcal{R}and(\mathcal{N},\mu)$ of $m = 1000$ random datasets was generated to represent the range of possible mRNA fluctuations consistent with the EWS-FLI1 perturbation.

From our analyses we can see that when using the generic influence graph we obtained a consistency P-value of 52.4% (524 random consistent datasets over 1000), whereas with the refined graph we obtained a consistency P-value of 31.2% (312 random consistent datasets). This confirms that the generic constraint used to model the generic influence graph does not constrain enough gene fluctuations. Furthermore, adding logical rules only on 36 nodes (10% of the total number of nodes) of biological interest, highly reduces the range of possible observations compatible with the topology of the model. In other words, the refined influence graph is more specific than the generic.

Finally, we computed the prediction probabilities, that is the chance of obtaining the same predictions when the models are confronted with random datasets (see Table 6.6). The prediction probability was 16.3% on average with the generic influence graph. It decreased to 9.9% with the refined graph. The nodes with low prediction probability express that their $\{+,-\}$ value is not easy to predict by chance, being specific to the EWS-FLI1 signals.

Biological interpretation of the results 6.5. *Designing logical rules allows us to significantly reduce the range of possible observations that are consistent with the initial observation dataset. Also, it generates non-trivial predictions with low prediction probability. The prediction probabilities are in the trivial case reduced to the probability of choosing an specific mRNA by chance. This is the case of all the predictions obtained with the generic influence graph. By adding logical rules in the model, we modify the distribution of their prediction probabilities as shown in Table 6.6.*

6.2.5 Studying the correlation between the cell cycle S-phase progression and the EWS-FLI1 activation

The new qualitative functions added into the model increased the number of $\{+,-\}$ predictions on the network nodes. This opened the way to investigate which signaling pathways can explain an observed phenotype. In Section 3.1.2.3 we proposed two programs that, when given a network $\mathcal{N}$ and a dataset μ of observations, obtain a subset of interactions of $\mathcal{N}$ and a subset of observations of μ that minimally explain a known fact. In our case we were interested on uderstanding:

(i) the arrest of cell cycle when EWS-FLI1 is inactivated, and

(ii) the restarting of cell cycle when EWS-FLI1 is reactivated.

On that account we confronted the refined influence graph with datasets A and B. The output of both programs, shown as a graph, represents a cascade of interactions originated by a small set of genes. The combined behavior of these genes explained the known variation of the cell cycle progression.

Table 6.6: Probabilities of the predictions calculated using the generic and refined influence graphs. The predicted nodes are listed in descending order according to the prediction probabilities obtained with the refined model. The *Complex Function* column specifies which nodes were modeled using a complex regulatory function. The two last columns describe the topological degree of the predictions: *Regulators* lists how many and which type of regulators had the predicted node, while *Targets* how many and which type of the *unique* targets had the predicted node. By unique we refer to those targets having as only predecessor the predicted node. The '*' symbol means multiple.

Predicted node	Generic Model Prediction Prob.	Refined Model Prediction Prob.	Complex Function	Regulators	Targets (only regulator)
ANAPC11_ANAPC2_CDC20 → −	52,3	31,1		*	1 (S_F)
TGFB → +	52,3	31,1		*	1 (S_F)
ANAPC11_ANAPC2_FZR1 → −	52,3	31,1		*	1 (S_F)
E2F8 → −	24,9	14,4		1 (mRNA)	
E2F1 → −	24,9	14,4		1 (mRNA)	
E2F6 → −	24,9	14,4		1 (mRNA)	
CCND3 → −	24,9	14,4	X	1 (mRNA)	
mRNA_E2F6 → −	15,2	9,5		1 (protein)	
JUN → +	11,4	9		*	1 (mRNA)
mRNA_E2F8 → −	15,1	9		1 (protein)	
CCND2 → −	12,3	8,6	X	1 (mRNA)	
MYC → −	12,3	8,6		*	1 (mRNA)
CDC2 → −	13	8,3		1 (mRNA)	
PTPN11 → −	11,3	8,1		1 (mRNA)	
TNFSF18 → +	12,9	8		1 (mRNA)	
IER3 → +	10	7,9		1 (mRNA)	
CCNB1 → −	11,6	7,7		1 (mRNA)	
TNFAIP3 → +	10,4	7,5		1 (mRNA)	
TNFSF15 → +	11	7,5		1 (mRNA)	
CDKN2C → −	10,4	7,5	X	1 (mRNA)	
CDK6 → +	11,9	7,4		1 (mRNA)	
CDK2 → −	11,3	7,3		1 (mRNA)	
RASA1 → +	10,2	7,2	X	1 (mRNA)	
CCNA2 → −	14,1	7,2	X	1 (mRNA)	
MYCBP → −	10,4	7,1		1 (mRNA)	
SKP2 → −	13,2	7,1		1 (mRNA)	
PDGFB → +	11,6	7		1 (mRNA)	
CCND1 → −	13,2	6,9	X	1 (mRNA)	
ECM → +	10,4	6,9	X	1 (mRNA)	
PRKCB1 → −	12,9	6,6	X	1 (mRNA)	
CFLAR → +	10,3	6,5		1 (mRNA)	
NFKB → +	10,3	6,5		*	1 (mRNA)
BTRC → −	12,8	6,2		1 (mRNA)	
CCNH → −	12,7	6	X	1 (mRNA)	
CDK4 → −	12,9	5,8		1 (mRNA)	
RBL1 → −	0,11	no prediction		*	
CCNA2_CDC2_act → −	no prediction	0,187	X	*	
CCNA1_CDC2_act → −	no prediction	0,181	X	*	
CCNB1_CDC2_a → −	no prediction	0,17	X	*	
CCND3_CDK4_a → −	no prediction	0,144	X	*	
CCND3_CDK6_a → −	no prediction	0,144	X	*	
CCNA1_CDK2_act → −	no prediction	0,142	X	*	
CCNA2_CDK2_act → −	no prediction	0,139	X	*	
E2F2 → −	no prediction	0,127	X	1 (mRNA)	
CCND2_CDK4_a → −	no prediction	0,086	X	*	
CCND2_CDK6_a → −	no prediction	0,086	X	*	
E2F3 → −	no prediction	0,077	X	1 (mRNA)	
CCNB1_CDC2 → −	no prediction	0,077	X	*	
E2F5 → −	no prediction	0,072	X	1 (mRNA)	
CCND1_CDK6_a → −	no prediction	0,069	X	*	
CCND1_CDK4_a → −	no prediction	0,069	X	*	
mRNA_TP73 → −	no prediction	0,002		2 (proteins)	
mRNA_APAF1 → −	no prediction	0,002		2 (proteins)	
mRNA_CDKN2A → −	no prediction	0,001		2 (proteins)	
mRNA_RB1 → −	no prediction	0		2 (proteins)	

Phenotype I: the cell cycle S progression decreases when EWS-FLI1 is inhibited (dataset A)

The cell cycle S-phase progression ('ccS') node receives 9 influences in the refined influence graph of the EWS-FLI1 signaling network. We may shortly describe them as follows:

- Three influences are issued from the competition between proteins E2F1,2,3 and RB1. RB1 is a member of the pocket protein family known to sequestrate E2F1,2,3

when active and thus preventing the E2F1,2,3 normally transcription of genes important to the S-phase progression [DeG02].

- One influence is triggered by the CCNE-CDK2 active protein complex [OS02].
- Three influences are coming from E2F6,7,8, which were reported to inactivate the transcription of genes responsible for the S-phase progression [DJ06].
- Two influences are triggered from the complex formed by proteins E2F4,5 with pocket proteins RBL2,1, respectively [DeG02, OS02].

The 'ccS' node is modeled using the generic influence constraint (GEN_{ccS}), since we did not established the priority order of the 'ccS' regulators. However, experiments show that 'ccS' is inhibited when EWS-FLI1 is. In addition, in dataset A 54 network mRNA nodes were measured to change significantly due to EWS-FLI1 inhibition.

Our objective was to explain an important effect of the EWS-FLI1 inhibition: the decreasing in cell cycle S phase transition. In symbolic language, this can be viewed as extracting the subgraph linking observed mRNA signs with the 'ccS' node coded as '−'. We used the program described in Algorithm 4 to recover the subgraph depicted in Fig. 6.2. The results of this first analysis show that 3 of the 9 influences that 'ccS' receives are '−', 3 are '+' (thus contrary to the 'ccS' observation), and 3 are not fixed, that is, they may be '+' or '−' without representing a contradiction between the model and the dataset.

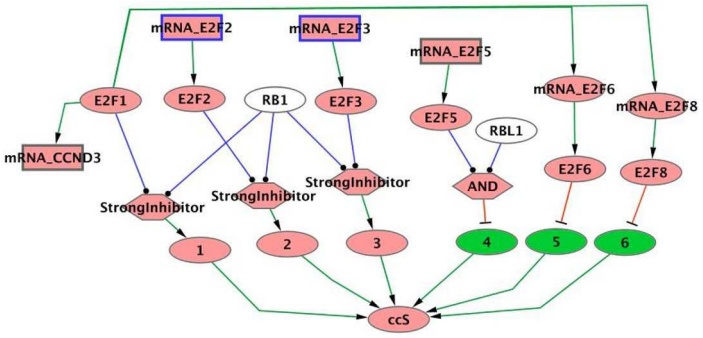


Figure 6.2: EWS-FLI1 network subgraph describing some of the influences that the cell cycle node ('ccS') receives. Arrows ending in '–>' or '–|' refer to activations or inhibitions, green/red nodes represent up/down-regulated products, and octagonal nodes are those modeled using non-generic constraints. The depicted 'ccS' influences (nodes 1-6) were predicted during the EWS-FLI1 inhibition. The observations in dataset A causing these fixed influences were tracked down (rectangular nodes).

In Fig. 6.2 we describe the three inhibited pathways when EWS-FLI1 is inhibited. Notice that there is not a direct connexion between EWS-FLI1 and the 'ccS' inhibition. Hence, the propagation of EWS-FLI1 inhibition can be tracked down. Its effect on cell cycle S transition is mediated via the negative fluctuation of three of the nodes in the network: $E2F2^{mRNA}$, $E2F3^{mRNA}$, and $CCND3^{mRNA}$. Interestingly, the first

two nodes do not receive other influences in the network. The three influences that contradict the sense of fluctuation of the S-phase progression are also depicted in Fig. 6.2. One of them comes from the protein complex formed by E2F5 and RBL1, which is reviewed to inactivate the S-phase progression in [DeG02]. In this review they also illustrate a case were E2F5 -/- Mefs mutant cells show cell cycle arrest, even after the E2F5-RBL1 complex is not formed. We arrive to a similar conclusion since the influence coming from E2F5-RBL1 complex does not impact the arrest of the cell cycle progression. The other two influences come from E2F6 and E2F8. Literature is still elusive on the role of those genes. In the absence of further knowledge, these results suggest that in the case of A673 cell lines, those influences are not enough to promote the cell cycle progression.

Biological interpretation of the results 6.6. *E2F2 and E2F3 transcriptions may be regulated directly or indirectly by EWS-FLI1. This regulation may be enough to stop A673 cells in G1 phase.*

Phenotype II: the cell cycle S progression activates when EWS-FLI1 reactivates (dataset B)

We showed that the specific rules introduced into the model were able to explain the 'ccS' inactivation when EWS-FLI1 is inhibited. However, when applying Algorithm 4 to the dataset B, it was not possible to conclude which influence could explain the cell cycle progression ('ccS' node observed as '+'). This is a consequence of our generic modeling, which does not allow us to predict the fluctuation of a node when opposite-signed influences arrive to it.

To overcome this problem we used the method described in Algorithm 5, that answers to the following question: Which network (non-observed, non-predicted) product has to be fixed to provide a path of influences that explains a positive change of the 'ccS' node without contradicting the known observations in dataset B? The subgraph outputted from Algorithm 5 is shown in Fig. 6.3. The new nodes predicted to have a $\{+,-\}$ variation in order to explain the '+' variation of the 'ccS' node are shown in Fig. 6.3 in dark green/red colors.

We extracted a subset of these new predictions composed of nodes which variation is enough to explain at least one of the influences over 'ccS'. These nodes are: $(RB1,-)$, $(RBL1,2,-)$, $(E2F7^{mRNA},-)$, and $(CCNE^{mRNA},+)$. As a further step, we applied Algorithm 4 to these nodes, to check if the $\{+,-\}$ predicted value could have an origin in some of their observed predecessors. The results are shown in Table 6.7. The $E2F7^{mRNA}$ illustrates a clear example of priority order. In order to provide a positive influence over the 'ccS' node, $E2F7^{mRNA}$ needs to be repressed by E2F4. The second influence $E2F7^{mRNA}$ receives, coming from E2F1, should be absent or not strong enough.

Biological interpretation of the results 6.7. *The pocket family proteins (RB1, RBL1, RBL2) should be inactivated in order to explain the 'ccS' node as activated (+). The RBL1 mRNA is observed as '+' in dataset B, while the mRNAs of the remaining*

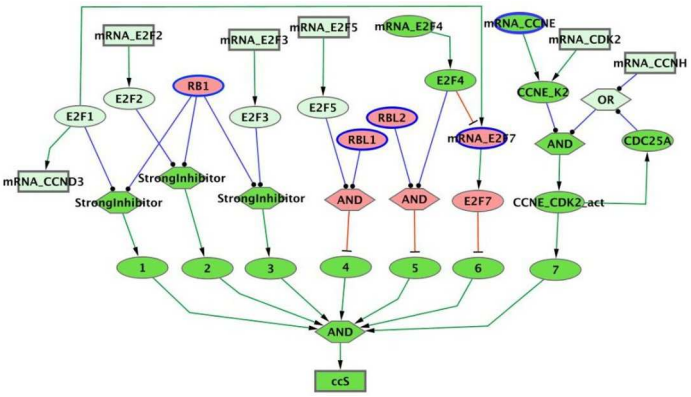


Figure 6.3: Predicted $\{+,-\}$ variations (green/red colors) in order to provide 7 positive influences over the 'ccS' node. Arrows ending in '->' or '-|' refer to activations or inhibitions, octagonal-shaped nodes are those modeled using logical functions. The impact of the dataset B observations (rectangular nodes) on the pathways controlling the 'ccS' node is given by the light-green colored nodes. This impact cannot predict as '+' any of the 7 influences that the 'ccS' receives. By adding the AND_{ccS} function into the model, new predictions (red/dark green nodes) were generated consistent with the dataset. These new predictions justified the '+' change of the 'ccS' node. A blue-bordered node is enough to explain at least one influence over the 'ccS' (nodes 1-7) as '+'.

Table 6.7: Initially non-observed/non-predicted products that, if fixed to a specific sign (in parenthesis), explain at least one positive influence over 'ccS' as '+'. The upper 'e' in the name of the product specifies its mRNA variation. *NR* is the number of regulators of the product, *E vs. O* shows how many regulators explain the sign of the product (E) vs. how many the opposite sign (O), and *Origin E/O* is the set of mRNA observations in the dataset that explain/contradict the sign of the product. When only one origin is found, we list in addition the name and role of the regulator that propagated the effect over the product.

Product	NR	E vs. O	Origin E	Origin O
RB1 (−)	11	7 vs. 0	CCND1,2,3 (+) PRKCB (+) CDK4 (+) CCNH (+)	
RBL1,2 (−)	10	6 vs. 0	CCND1,2,3 (+) CDK4 (+) CCNH (+)	
$E2F7^e$ (−)	2	1 vs. 1	E2F4 (+) E2F4 -|	CCND3 (+) E2F1 ->
$CCNE^e$ (+)	7	5 vs. 1	E2F2,3 (+) CCND2,3 (+) EWS-FLI1 (+)	CCND3 (+) E2F8 -|

pocket proteins are predicted also to '+'. This means that during the EWS-FLI1 reactivation the mRNA of the pocket proteins increase. Therefore, these proteins should be inactivated due to post-translational phosphorylations coming from any of the products listed in the column "Origin E" in Table 6.7.

Biological interpretation of the results 6.8. *One possible pathway that activates the cell cycle S-phase may come from the $CCNE^{mRNA}$ node. This node has 7 regulators in the network; 5 of them provide an explanation for a '+' change over it, including EWS-FLI1. Only one of its regulators contradicts this change, E2F8, which also in Fig. 6.2 appeared to contradict the 'ccS' expected behavior.*

6.2.6 Discussion

We showed in these last sections how we conceived and tested new qualitative functions to model the effect of complex regulatory influences on steady state changes in a signaling network. Based on a combination of the generic function GEN and of Boolean operators among signs $\vee$ and $\wedge$, we designed functions to model the global response to a stress of various macromolecular interactions such as competitions, sequestration and releasing, and complex inactivation-reactivation. In comparison with a model that does not implement such functions, we show that the prediction number is significantly increased, both on experimental data and on random datasets. Experimentalists can have more confidence in those predictions: using new functions decrease significantly the probability to obtain them on random datasets. Moreover, our predictions concern active proteins, difficult to be widely measured by high-throughput methodologies.

We proposed a methodology that exploits those new functions to study the impact of targeted manipulations on key nodes in the network. Biologically, these *in silico* manipulations can simulate the effect of a manipulation on target genes, using inhibitors (drugs or shRNA) or enhancers (*e.g.* strong promoters). Doing this we assume that the globally observed variations on the network nodes remain stable when one manipulates specific molecules. With the methodology proposed one can work on networks relatively large (hundreds of products), for which one cannot manage to reason in an intuitive way.

The presented methods generate new hypotheses that suggest new experiments. First, on the experimental dataset we were able to make predictions on the behavior of specific nodes such as E2F5, IER3, or the PDGF pathway; these predictions should be confirmed experimentally. Second, the *in silico* manipulation of nodes influencing the cell cycle transition to phase S suggests that E2F2,3 are direct or indirect targets of EWS-FLI1; also, a list of variations of some network nodes is predicted to be at the origin of the cell cycle reactivation. Our analysis points out different network nodes important in the cell cycle S phase transition, that could be interesting targets for biologists in order to better understand the correlation between EWS-FLI1 and the cell cycle.

Conclusion

The work in this thesis was centered on the study of large-scale biological regulatory networks. Having in mind that these systems are composed of thousands of regulations, we used a qualitative approach in which a regulatory network was represented as a set of qualitative constraints. Afterwards, we reasoned over the steady state equilibrium shifts of the network molecules by checking for each network component a generic consistency rule. This intuitive reasoning is relevant and different from other approaches because it is automatic and global: it is computed over the whole network topology.

Efficient algorithms were developed to propose an informatic solution of this problem in reasonable computational time. Using them, we presented in this thesis programs implemented in Python and ASP, as well as bioinformatic tools, such as a Website, a visualization tool, and a Web service, which automatize the consistency analysis and are publicly available to be tested on different type of biological data.

The consistency analysis proposed needs as biological data input a regulatory network and a dataset of qualitative shifts of network molecules. We acquired this data from public databases, literature, and from the collaboration with other research projects. In all the cases a post-modeling of the proposed network was performed. In the simplest case we filtered the original interactions with a threshold. In the more complex cases, we added new regulatory rules into the model by a careful reading of the literature. The expression datasets were also treated in order to be analyzed using the qualitative approach.

The results of this thesis concern firstly the biological validation of the consistency approach, and secondly the mathematical and informatic extension of this approach in order to broaden the types of experimental datasets considered and to represent more specific regulatory interactions. We will summarize the contributions of this thesis, and give a final perspective of this work in the following sections.

Consistency-check of signed transcriptional networks

The consistency-check analysis is a process aimed to confront a regulatory network and a dataset of observations. Three steps are always performed to compute this analysis:

1. Check the consistency answer of the confrontation between regulatory and expression data.

2. If the data is not consistent, diagnose the network regulations and dataset observations involved in the inconsistency.

3. If the data is consistent, predict the qualitative shift of some network molecules.

We applied this process to the *E. coli* TRN (extracted from RegulonDB) wrt a small dataset of reliable but heterogeneous *E. coli* observations on the exponential-stationary growth shift. Initially both data were inconsistent. The inconsistencies highlighted in the diagnosis step, together with a careful review of the literature on the nodes and regulations involved in the inconsistencies, revealed that:

- The sigma-factors regulations needed to be added to the transcriptional *E. coli* regulations in order to obtain a model that reflects equilibrium shifts.
- The '$ihfA = -$' observation, reported by the literature-curated dataset of the exponential-stationary growth shift that appears in RegulonDB, is inconsistent with the transcriptional and sigma-factors regulations.
- The regulation '`ArcA -> appY`' was missing in the transcriptional plus sigma-factors model proposed by RegulonDB.

Once we obtained a consistent regulatory model with the experimental dataset, we predicted the variation of some network molecules. We noticed that by adding an additional constraint into the model, obtained from quantitative information related to the formation of the IHF protein-complex, we increased significantly our prediction rate by 30%. After validating the predicted values by using a genome-wide dataset of mRNA measurements, we obtained an 80% of accuracy. This percentage is comparable to the one obtained by other methods working on a metabolic and more complex *E.coli* regulatory model [CKR$^+$04, CP02, EP00].

Our results also suggested that the *E. coli* TRN can be completed with post-translational interactions by a careful (literature or experimental) study of the non-validated predictions. We illustrated this idea in the prediction of the RpoD protein. We verified in the literature that its predicted value referred to the active protein, and thus it did not correlate with its mRNA level reported in the genome-wide dataset.

The *E. coli* TRN was also confronted wrt genome-wide datasets. In this case, as the number of inconsistencies reported is higher, we cannot afford checking all of them manually. Nevertheless, global properties of these datasets can be proposed, according to the number of recovered predictions after an automatic repair.

We studied two genome-wide datasets related to the exponential-stationary growth shift of *E. coli* genes and to the Heatshock stress perturbation. The predictions of these datasets, after minimal correction of the inconsistencies, were highly accurate (90%). This implies that the minimal nodes in the network to be corrected may be strong candidates for experimental validations. We computed these candidates for the growth shift dataset obtaining 11 molecules, which observed value in the dataset or input regulations needed to be corrected in order to reconcile the network and dataset.

TF role inference on unsigned transcriptional networks

The TF role inference approach consists of predicting the $\{+, -\}$ role of a transcription factor over a gene based on a consensus of observations reported by multiple gene expression profiles. This consensus should be consistent with the network topology. We tested this approach on the *E. coli* and *S. cerevisiae* TRNs.

Based on the unsigned *E. coli* TRN we simulated random consistent datasets. These simulations allowed us to explore how many experiments were needed to infer all the regulatory roles. Our results revealed that the number of predicted roles were dependant on the network topology. For highly connected networks, such as the core of a network, we predicted less regulatory roles, even when observing all the network molecules artificially. Our simulations showed that one can characterize about 30% of the regulations from 30 expression profiles. We additionally showed that this is close to the theoretical limit of our approach.

After applying the inference approach on real biological data, we obtained much less number of inconsistencies when using the *E. coli* regulatory model. The causes of this difference may be the higher number of non-modeled post-transcriptional interactions present in eukaryotic networks. A second cause is that the RegulonDB regulatory model of *E. coli* is better curated and more reliable than the model obtained from ChIP-chip data of *S. cerevisiae*.

We compared our inference approach to the path analysis method of Yeang and colleagues [YIJ04, YMM$^+$05] and found that both algorithms infer a similar number of regulations, and that the common predictions agreed. We noticed that the predictions are located in different parts of the network, depending on the algorithm: path analysis tends to infer signs in highly connected regions, while our approach infer signs on regulations acting on small in-degree nodes. Another difference is that path analysis requires expression profiles from gene-deletion experiments, whereas our method gives better results with stress perturbation experiments (though it can be applied to both types of experiments).

The TF role predictions obtained from *E. coli* (using 61 datasets) and *S. cerevisiae* (using 15 datasets) were validated using literature curated signed regulatory models. The prediction rate for *E. coli* was of 30%, and 71% of them were accurate. The prediction rate for the largest *S. cerevisiae* model was of 61%, and 69% of them were accurate. Nevertheless, only 23 of the 631 predicted roles for *S. cerevisiae* could be validated, since the only source of comparison found was a small-scale signed regulatory network.

Consistency-check on signed signaling networks

To model the post-translational phenomena present in the signaling network of the EWS-FLI1 oncogene, we added specific qualitative constraints in the model. These constraints were built according to the biological literature by using a combination of the Boolean operators $\vee$ and $\wedge$ among $\{+, -\}$ signs. Hence, the mathematical and

informatic approaches needed to be adapted to reason over this new qualitative system of constraints.

We validated the new logical rules added into the model by comparing a network modeled with specific logical rules wrt a network modeled with only generic constraints. Our results revealed that the rate of predictions was increased by 44.7% owing to the additional specific rules. Also, the predictions generated with the more specific model were more significant, that is, it was less probable to obtain these predictions from a random dataset.

Using the EWS-FLI1 signaling network we went one step further from our previous analyses. We performed a post-analysis of the predictions obtained in order to find the origin of observed phenotypes. In the EWS-FLI1 signaling network, EWS-FLI1 is known to correlate with the cell-cycle activation. Thus, we studied which nodes in the network may be responsible for the cell-cycle inhibition and reactivation. We obtained three interesting hypothesis to be validated with further experimental manipulations:

(i) *Completing the signaling network.* E2F2 and E2F3 transcriptions may be regulated directly or indirectly by EWS-FLI1. These regulations may be enough to stop the Ewing tumour A673 cell line in G1 phase.

(ii) *Explaining the cell-cycle reactivation.* The pocket family proteins (RB1, RBL1, RBL2) should be inactivated in order to explain the reactivation of the cell-cycle. The mRNA of RBL1 is observed in the dataset to increase, and the others are predicted also to increase. Thus, these proteins should be inactivated due to post-translational phosphorylations. We reported the list of products responsible of these phosphorylations.

(iii) *Explaining the cell-cycle reactivation.* Another cause of the cell-cycle reactivation lies on the increase in expression of the CCNE mRNA. The causes of this increase are the shift in expression of five network molecules.

Perspectives

Refining and Curating a model of regulations

The approach proposed in this thesis can be applied to construct or validate regulatory models for organisms whose regulatory map is not yet built. Let us take for example the work with the *Acidithiobacillus ferrooxidans* bacteria [RGKS03]. Even though the complete genome of this bacteria was sequenced, the information of its genome is still not complete. In addition, it is difficult to generate a regulatory model for it from mutants data because this bacteria is highly resistant to experimental alterations of its genome. Nonetheless, the following data is available:

- *Regulatory data.* This data includes the complete annotation of the bacteria genome, a list of important transcription factors in this bacteria, and a compilation of possible TFs that share a binding site up-stream all the bacteria genes.

- *Experimental data.* 13 genome-wide microarray measurements on this bacteria under different stresses.

By using different filters of the regulatory data, we may build several putative models of regulation. Thus, we could iteratively test the consistency of these models wrt all the experimental profiles, in order to obtain a closer map of the transcriptional regulations of this organism. A ranking of the putative networks could also be computed by finding the number of minimal repairs of the networks wrt all the experimental datasets.

Distance between a network and genome-wide datasets

The notion of a distance between experimental and regulatory data is another interesting direction of this work. We arrived very close to it by using the minimal automatic correction programs in ASP. Nevertheless, the biological impact of this work could be better achieved by proposing a number that measures the degree of consistency of different genome-wide datasets wrt the same regulatory model. This idea is connected with the example given for the *Acidithiobacillus ferrooxidans* regulatory network and it may have an interesting impact in the correction and validation of regulatory models.

In a previous work using the *E. coli* network and three genome-wide datasets, we classified small sets of molecules of this network into two groups. These groups were obtained by randomly observing small sets of genes in the network. The first group of genes led to a high prediction rate, whereas the other to a low prediction rate. It should be interesting to go further in this direction and study the notion of consistency from a topological point of view. We already know that not all the nodes in the network have the same relevance. However, the simulation of random groups of observed genes may give us insights on the key groups of genes essential to control the cell response of many others. It may be possible that other TFs, different from the global factors, are responsible of this high control.

The advantage of applying the distance notion to analyze wide-genome datasets is that the informatic tools are already available. It just requires a further research on the steps of this analysis, as well as an interpretation of the obtained results.

Further improvements of the consistency analysis

The Bioquali library was extended in order to represent the null-variation of some network products. In general terms, it is done by splitting all the nodes of the graph in two values `(presence,variation)`: *presence*, is a Boolean value in $\{0, 1\}$ that indicates the absence or presence of a molecule, while *variation* indicates the $\{+, -, ?\}$ variation of a molecule. If a molecule is represented as `(0,?)`, it means that it has not changed between the two experimental conditions we compare; if it is represented as `(1,+)`, then it was up-regulated. Consequently, the $\oplus$ and $\otimes$ sign operators, used to compute the consistency of a constraint with the generic rule, evolve as shown in Table 6.8.

The drawback of this improvement of the Bioquali library is that each time a consistency analysis is launched, the variables of the system are doubled. Thus, the computation time of an automatic consistency analysis is unbearable in its actual state.

Table 6.8: Consistency relation $\simeq$ and sign tables for the addition ($\oplus$) and multiplication ($\otimes$) extended taking into account the null-variation (0 change) of a product. The $\simeq$ relation states the consistency answer of each constraint, T stands for true, whereas F for false. The 'x' represents that this never happens.

$\simeq$	$+$	$-$	0	**?**
$+$	T	F	F	T
$-$	F	T	F	T
0	F	F	T	T
?	T	T	T	T

$\oplus$	$+$	$-$	0	**?**
$+$	$+$	**?**	$+$	**?**
$-$	**?**	$-$	$-$	**?**
0	$+$	$-$	0	**?**
?	**?**	**?**	**?**	**?**

$\otimes$	$+$	$-$	0	**?**
$+$	$+$	$-$	0	**?**
$-$	$-$	$+$	0	**?**
0	0	0	x	0
?	**?**	**?**	0	**?**

Validating the impact of the null-variation over the prediction rate or accuracy could be another nice direction to pursue this work. This will require improving the existing Bioquali methods, as well as a careful modeling study of the significance of the null-variation in terms of an equilibrium shift. The ASP approach can also be easily extended in this direction, however, before implementing it, it is important to impose a correct modeling of the null-variation.

A recurrent idea in the consistency analysis process is the experimental planning design, for example, controlling the network of interactions in a way to predict a $\{+,-\}$ change by switching *off* or *on* only few molecules in the network. As we showed with the EWS-FLI1 signaling network, this result is strictly related to the number of fixed (predicted) network molecules and the generic consistency rule may cause ambiguities when a node has more than one regulator. This direction is promising and very interesting from the mathematical, informatics, and biological perspectives. It is crucial, however, that methodological changes are made in close collaboration with experimentalists and biologists. The validation of the computational output needs to be considered by a biological laboratory working on the same problematic. If not, we risk to make interesting (and costly) contributions in the informatics and mathematics fields without any application in the biological side.

The Systems Biology field is a research domain with extraordinary directions. It is impressing to see how by mathematically and computationally analyzing high-throughput data, small models of regulatory interactions can be elucidated which reflect experimental observations. The idea of predicting the behavior of biological cell processes is promising. However, most of these researches are centered on small regulatory networks, connecting either very well studied transcription factors or top-ranked measured molecules. It is a promising direction to study and elucidate properties in network models of hundreds or thousands of interactions, since this means getting closer to the biological reality. Biological organisms are too complex and have evolved in a so large time scale, that even their basic processes have to be composed of plenty intra-molecular messages. Moreover, nowadays when experimental technology allows us to explore thousands of molecules at once, and the informatic facilities are able to process large amount of information, it seems reasonable (but still challenging) to explore the large-scale di-

rection concerning network regulatory models. With tools as the consistency analysis presented in this thesis we did our small contribution to the research in this direction.

Bibliography

[AAIN+91] T Ali Azam, A Iwata, A Nishimura, S Ueda, and A Ishihama. Growth Phase-Dependent Variation in Protein Composition of the Escherichia coli Nucleoid. *J Bacteriol*, 181(20):6361–6370, August 1991.

[ABC99] M. Arenas, L. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'99)*, pages 68–79. ACM Press, 1999.

[AGS+94] M Aviv, H Giladi, G Schreiber, AB Oppenheim, and G Glaser. Expression of the genes coding for the Escherichia coli integration host factor are controlled by growth phase, rpos, ppgpp and by autoregulation. *Mol Microbiol*, 14(5):1021–31, Dec 1994.

[AHL+03] TE Allen, MJ Herrgård, M Liu, Y Qiu, JD Glasner, FR Blattner, and BØ Palsson. Genome-scale analysis of the uses of the Escherichia coli genome: model-driven analysis of heterogeneous data sets. *J Bacteriol*, 185(21):6392–9, Nov 2003.

[ARS+08] Y. Assenov, F. Ramírez, S. E. Schelhorn, T. Lengauer, and M. Albrecht. Computing topological parameters of biological networks. *Bioinformatics*, 24:282–284, Jan 2008.

[ASOB96] T Atlung, S Sund, K Olesen, and L Brøndsted. The histone-like protein H-NS acts as a transcriptional repressor for expression of the anaerobic and growth phase activator AppY of Escherichia coli. *J Bacteriol*, 178(12):3418–3425, Jun 1996.

[asp] Reasoning on biological models using asp. http://www.cs.uni-potsdam.de/wv/bioasp.

[B06] Palsson B. *Systems Biology - Properties of Reconstructed Networks*. Cambridge University Press, 2006.

[BA96] L Brondsted and T Atlung. Effect of growth conditions on expression of the acid phosphatase (cyx-appa) operon and the appy gene, which

encodes a transcriptional activator of Escherichia coli. *J Bacteriol*, 178(6), Mar 1996.

[BA08] Jan Baumbach and Leonard Apeltsin. Linking Cytoscape and the corynebacterial reference database CoryneRegNet. *BMC Genomics*, 9:184, 2008.

[Bar03] C. Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, 2003.

[BB99] PO Brown and D Botstein. Exploring the new world of the genome with DNA microarrays. *Nat Genet*, 21:33–7, 1999.

[BBAIdB07] M Bansal, V Belcastro, A Ambesi-Impiombato, and D. di Bernardo. How to infer gene networks from expression profiles. *Mol Syst Biol.*, 3(78), 2007.

[BDGdB06] M. Bansal, G. Della Gatta, and D. di Bernardo. Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, 22:815–822, 2006.

[bio] Bioquali cytoscape plugin java web start. genoweb.irisa.fr/Serveur-GPO/outils/interactionNetwork/BIOQUALI/BioqCyPlugin/JWS/BioqCyPlugin.jnlp.

[BMA+07] MD Bradley, Beach MB, De Koning AP, Pratt TS, and Osuna R. Effects of Fis on Escherichia coli gene expression during different growth stages. *Microbiology*, 153:2922–40, 2007.

[Bor09] M Le Borgne. Solving losely coupled constraints. Technical Report RR-6958, INRIA, 2009.

[BRdJ+05] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider. Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in Escherichia coli. *Bioinformatics*, 21 Suppl 1:19–28, Jun 2005.

[BrPB+97] FR Blattner, G 3rd Plunkett, CA Bloch, NT Perna, V Burland, and M Riley. The complete genome sequence of Escherichia coli K-12. *Science*, 277(5331):1453–74, Sep 1997.

[Bry86] R. Bryant. Graph-based algorithm for boolean function maipulation. *IEEE Transactions on Computers*, 35(8):677–691, 1986.

[BTR09] Jan Baumbach, Andreas Tauch, and Sven Rahmann. Towards the integrated analysis, visualization and reconstruction of microbial gene regulatory networks. *Brief Bioinform*, 10(1):75–83, 2009.

[BWR+07] Jan Baumbach, T Wittkop, K Rademacher, S Rahmann, K Brinkrolf, and A Tauch. CoryneRegNet 3.0–an interactive systems biology platform for the analysis of gene regulatory networks in corynebacteria and Escherichia coli. *J Biotechnol.*, 129(2):279–289, 2007.

[CCG+02] Alessandro Cimatti, Edmund M. Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In *Computer Aided Verification, 14th International Conference*, volume 2404 of *Lecture Notes in Computer Science*, pages 359–364. Springer, 2002.

[CCRFS06] Laurence Calzone, Nathalie Chabrier-Rivier, François Fages, and Sylvain Soliman. Machine learning biochemical networks from temporal logic properties. In *T. Comp. Sys. Biology*, volume 4220 of *Lecture Notes in Computer Science*, pages 68–94. Springer, 2006.

[CDE+98] S Chu, J DeRisi, M Eisen, J Mulholland, D Botstein, P O Brown, and I Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, 282(5389):699–705, 1998.

[CFS06] L. Calzone, F. Fages, and S. Soliman. BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22:1805–1807, Jul 2006.

[CHG+06] C Constantinidou, JL Hobman, L Griffiths, MD Patel, CW Penn, JA Cole, and TW Overton. A reassessment of the FNR regulon and transcriptomic analysis of the effects of nitrate, nitrite, NarXL, and NarQP as Escherichia coli K12 adapts from aerobic to anaerobic growth. *J Biol Chem*, 281(8):4802–15, 2006.

[CKR+04] MW Covert, EM Knight, JL Reed, MJ Herrgard, and BO Palsson. Integrating high-throughput and computational data elucidates bacterial networks. *Nature*, 429(6987), 2004.

[CLL+07] H. Y. Chuang, E. Lee, Y. T. Liu, D. Lee, and T. Ideker. Network-based classification of breast cancer metastasis. *Mol. Syst. Biol.*, 3:140, 2007.

[CP02] MW Covert and BO Palsson. Transcriptional regulation in constraints-based metabolic models of Escherichia coli. *J biol chem*, 277(31), 2002.

[CRCD+04] N Chabrier-Rivier, M Chiaverini, V Danos, F Fages, and V Schächter. Modeling and querying biochemical interaction networks. *Theoretical Computer Science*, 325:25–44, 2004.

[CSC+07] M. S. Cline, M. Smoot, E. Cerami, A. Kuchinsky, N. Landys, C. Workman, R. Christmas, I. Avila-Campilo, M. Creech, B. Gross, K. Hanspers, R. Isserlin, R. Kelley, S. Killcoyne, S. Lotia, S. Maere, J. Morris, K. Ono, V. Pavlovic, A. R. Pico, A. Vailaya, P. L. Wang, A. Adler, B. R. Conklin, L. Hood, M. Kuiper, C. Sander, I. Schmulevich, B. Schwikowski, G. J. Warner, T. Ideker, and G. D. Bader. Integration of biological networks and gene expression data using Cytoscape. *Nat Protoc*, 2:2366–2382, 2007.

[CSG+04] Paul J Cullen, Walid Jr Sabbagh, Ellie Graham, Molly M Irick, Erin K van Olden, Cassandra Neal, Jeffrey Delrow, Lee Bardwell, and George F Jr Sprague. A signaling mucin at the head of the Cdc42- and MAPK-dependent filamentous growth pathway in yeast. *Genes Dev*, 18(14):1695–708, 2004.

[CSP01] M W Covert, C H Schilling, and B Palsson. Regulation of gene expression in flux balance models of metabolism. *J Theor Biol*, 213(1):73–88, 2001.

[CVSM+09] J Collado-Vides, H Salgado, E Morett, S Gama-Castro, V Jiménez-Jacinto, I Martínez-Flores, A Medina-Rivera, L Muñiz-Rascado, M Peralta-Gil, and A Santos-Zavaleta. Bioinformatics resources for the study of gene regulation in bacteria. *J Bacteriol.*, 191(1):22–31, 2009.

[CXCK08] M. W. Covert, N. Xiao, T. J. Chen, and J. R. Karr. Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli. *Bioinformatics*, 24:2044–2050, Sep 2008.

[cyt] Cytoscape 2.x plugins. http://chianti.ucsd.edu/cyto_web/plugins/.

[DBTG+05] D. Di Bernardo, M. Thomson, T. Gardner, S. Chobot, E. Eastwood, A. Wojtovich, S. Elliot, S. Schaus, and J. Collins. Chemogenomic profiling on a genome-wide scale using reverse-engeneered gene networks. *Nat. Biotechnol.*, 23:377–383, 2005.

[DeG02] J DeGregory. The genetics of the E2F family of transcription factors: shared functions and unique roles. *Biochim Biophys Acta.*, 1602(2):131–50, 2002.

[DFF+07] V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine. Rule-based modelling of cellular signalling, invited paper. In L. Caires and V.T. Vasconcelos, editors, *Proceedings of the Eighteenth International Conference on Concurrency Theory, CONCUR '2007, Lisbon, Portugal*, volume 4703 of *Lecture Notes in Computer Science*, pages 17–41, Lisbon, Portugal, 3–8 September 2007. Springer, Berlin, Germany.

[DIB97] J L DeRisi, V R Iyer, and P O Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–6, 1997.

[DJ02] H De Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[DJ06] J DeGregori and DG Johnson. Distinct and Overlapping Roles for E2F Family Members in Transcription, Proliferation and Apoptosis. *Curr Mol Med*, 6(7):739–48, 2006.

[dJGHP03] H. de Jong, J. Geiselmann, C. Hernandez, and M. Page. Genetic Network Analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19:336–344, Feb 2003.

[DL04] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theor. Comput. Sci.*, 325(1):69–110, 2004.

[Dor88] JL Dormoy. Controlling qualitative resolution. In *7th National Conference on Artificial Intelligence, AAAI'88*, Saint Paul, Min., USA, 1988.

[DSB93] P Dersch, K Schmidt, and E Bremer. Synthesis of the Escherichia coli K-12 nucleoid-associated DNA-binding protein H-NS is subjected to growth-phase control and autoregulation. *Mol Microbiol.*, 8(5):875–89, May 1993.

[DSH+03] G Jr Dennis, BT Sherman, DA Hosack, J Yang, W Gao, HC Lane, and RA Lempicki. Vid: Database for annotation, visualization, and integrated discovery. *Genome Biol*, 4, Apr 2003.

[DTC+05] BL Drees, V Thorsson, GW Carter, AW Rives, MZ Raymond, I Avila-Campillo, P Shannon, and T Galitski. Derivation of genetic interaction networks from quantitative phenotype data. *Genome Biol.*, 6(4):R38, 2005.

[DZP+92] O Delattre, J Zucman, B Plougastel, C Desmaze, T Melot, M Peter, H Kovar, I Joubert, P de Jong, G Rouleau, et al. Gene fusion with an ets DNA-binding domain caused by chromosome translocation in human tumours. *Nature*, 359:162–165, 1992.

[EBK+08] J Ernst, Q K Beg, K A Kay, G Balázsi, Z N Oltvai, and Z Bar-Joseph. A Semi-Supervised Method for Predicting Transcription Factor-Gene Interactions in *escherichia coli*. *PLoS*, 3:e1000044, 2008.

[EP00] JS Edwards and BO Palsson. The Escherichia coli MG1655 in silico metabolic genotype: its definition, characteristics, and capabilities. *Proc Natl Acad Sci U S A*, 97(10):5528–33, May 2000.

[FDF+07] JJ Faith, ME Driscoll, VA Fusaro, EJ Cosgrove, B Hayete, FS Juhn, SJ Schneider, and TS Gardner. Many microbe microarrays database: uniformly normalized affymetrix compendia with structured experimental metadata. *Nucleic Acids Res*, Oct 2007.

[FDK+09] J Feret, V Danos, J Krivine, R Harmer, and W Fontana. Internal coarse-graining of molecular systems. In *Proc Natl Acad Sci*, volume 106, pages 6453–8, USA, 2009.

[FGAPTQCV08a] JA Freyre-González, JA Alonso-Pavón, LG Treviño-Quintanilla, and J Collado-Vides. Functional architecture of Escherichia coli: new insights provided by a natural decomposition approach. *Genome Biol.*, 9:10, 2008.

[FGAPTQCV08b] JA Freyre-González, JA Alonso-Pavón, LG Treviño-Quintanilla, and J Collado-Vides. Functional architecture of Escherichia coli: new insights provided by a natural decomposition approach. *Genome Biol*, 9(10):R154, 2008.

[FHT+07] Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner. Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol*, 5(1):e8, 2007.

[FLNP00] N Friedman, M Linial, I Nachman, and D Pe'er. Using Bayesian networks to analyze expression data. *J Comput Biol*, 7(3-4):601–20, 2000.

[FMKT04] Akira Funahashi, Mineo Morohashi, Hiroaki Kitano, and Naoki Tanimura. Celldesigner: a process diagram editor for gene-regulatory and biochemical networks. *Biosilico*, 1:159–162, 2004.

[FMS+07] F Ferrazzi, P Magni, L Sacchi, A Nuzzo, U Petrovic, and R Bellazzi. Inferring gene regulatory networks by integrating static and dynamic data. *Int J Med Inform*, Epub 2007 Sept 6, 2007.

[FSCR04] F Fages, S Soliman, and N Chabrier-Rivier. Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *J. Biol. Phys. Chem*, 4:64–73, 2004.

[GBMS09] C Guziolowski, A Bourdé, F Moreews, and A Siegel. BioQuali Cytoscape plugin: analysing the global consistency of regulatory networks. *BMC Genomics*, 26(10):244, 2009.

[GCJJPG+08] S Gama-Castro, V Jiménez-Jacinto, M Peralta-Gil, A Santos-Zavaleta, MI Peñaloza-Spinola, B Contreras-Moreira, J Segura-Salazar, L Muñiz-Rascado, I Martínez-Flores, H Salgado, C Bonavides-Martínez, C Abreu-Goodger, C Rodríguez-Penagos, J Miranda-Ríos, E Morett, E Merino, AM Huerta, and L Treviño-Quintanilla J Collado-Vides. RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic Acids Res.*, 36:D120–4, 2008.

[GCT08] A. González, C. Chaouiya, and D. Thieffry. Logical modelling of the role of the Hh pathway in the patterning of the Drosophila wing disc. *Bioinformatics*, 24:i234–240, Aug 2008.

[gen] Plateforme bioinformatique ouest-genopole. http://genoweb.irisa.fr.

[GGRS09] Carito Guziolowski, Jeremy Gruel, O Radulescu, and A Siegel. Curating a large-scale regulatory network by evaluating its consistency with expression datasets. In *Computational Intelligence Methods for Bioinformatics and Biostatistics - Selected revised papers*, volume 5488 of *Lecture Notes in Computer Science*, pages 144–155, Salerno, Italy, 2009. Springer-Verlag.

[GHM+01] A P Gasch, M Huang, S Metzner, D Botstein, S J Elledge, and P O Brown. Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. *Mol Biol Cell*, 12(10):2987–3003, 2001.

[GKNS07] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. clasp: A conflict-driven answer set solver. In C. Baral, G. Brewka, and J. Schlipf, editors, *Proceedings of the Ninth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'07)*, volume 4483 of *lnai*, pages 260–265. springer, 2007.

[GMBC+04] J. F. Guespin-Michel, G. Bernot, J. P. Comet, A. Mérieau, A. Richard, C. Hulen, and B. Polack. Epigenesis and dynamic similarity in two regulatory networks in Pseudomonas aeruginosa. *Acta Biotheor.*, 52:379–390, 2004.

[GN93] AE Granston and HA Nash. Characterization of a set of integration host factor mutants deficient for dna binding. *J Mol Biol*, 234(1):45–59, Nov 1993.

[GRRL+03] Rosa Maria Gutierrez-Rios, David A Rosenblueth, Jose Antonio Loza, Araceli M Huerta, Jeremy D Glasner, Fred R Blattner, and Julio Collado-Vides. Regulatory network of Escherichia coli: consistency between literature knowledge and microarray profiles. *Genome Res*, 13(11):2435–2443, 2003.

[GSK+00] A P Gasch, P T Spellman, C M Kao, O Carmel-Harel, M B Eisen, G Storz, D Botstein, and P O Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 11(12):4241–57, 2000.

[GST07] M. Gebser, T. Schaub, and S. Thiele. GrinGo: A new grounder for answer set programming. In C. Baral, G. Brewka, and J. Schlipf, editors, *Proceedings of the Ninth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'07)*, volume 4483, pages 266–271, 2007.

[GST+08] M. Gebser, T. Schaub, S. Thiele, B. Usadel, and P. Veber. Detecting inconsistencies in large biological networks with answer set programming. In *Proceedings of the Workshop on Constraint Based Methods for Bioinformatics (WCB'08)*, 2008.

[Guz06] C Guziolowski. Testing a new approach of Qualitative Modeling in Escherichia coli transcriptional network. Master's thesis, Université de Rennes 1, 2006.

[GVB+07] Carito Guziolowski, Philippe Veber, Michel Le Borgne, Ovidiu Radulescu, and Anne Siegel. Checking consistency between expression data and large scale regulatory networks: A case study. *Journal of Biological Physics and Chemistry*, 7:37–43, 2007.

[GW02] B Grunenfelder and EA Winzeler. Treasures and traps in genome-wide data sets: case examples from yeast. *Nat Rev Genet*, 3(9):653–661, Sep 2002.

[HBC+01] EL Hong, R Balakrishnan, KR Christie, MC Costanzo, SS Dwight, SR Engel, DG Fisk, JE Hirschman, MS Livstone, R Nash, R Oughtred, J Park, M Skrzypek, B Starr, R Andrada, G Binkley, Q Dong, BC Hitz, S Miyasato, M Schroeder, S Weng, ED Wong, KK Zhu, K Dolinski, D Botstein, and JM Cherry. Saccharomyces genome database. Avalaible: http://www.yeastgenome.org/, 2001.

[HCP03] MJ Herrgard, MW Covert, and BØ Palsson. Reconciling gene expression data with known genome-scale regulatory network structures. *Genome Res*, 13(11):2423–34, Nov 2003.

[Hec98] David Heckerman. A tutorial on learning with bayesian networks. In *Proceedings of the NATO Advanced Study Institute on Learning in graphical models*, pages 301–354, Norwell, MA, USA, 1998. Kluwer Academic Publishers.

[HFS+03] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden,

A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J. H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19:524–531, Mar 2003.

[HGL+04] Christopher T Harbison, D Benjamin Gordon, Tong Ihn Lee, Nicola J Rinaldi, Kenzie D Macisaac, Timothy W Danford, Nancy M Hannett, Jean-Bosco Tagne, David B Reynolds, Jane Yoo, Ezra G Jennings, Julia Zeitlinger, Dmitry K Pokholok, Manolis Kellis, P Alex Rolfe, Ken T Takusagawa, Eric S Lander, David K Gifford, Ernest Fraenkel, and Richard A Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, 2004.

[HLPP06] Markus J Herrgard, Baek-Seok Lee, Vasiliy Portnoy, and Bernhard O Palsson. Integrated analysis of regulatory and metabolic networks reveals novel regulatory mechanisms in Saccharomyces cerevisiae. *Genome Res*, 16(5):627–35, 2006.

[HMJ+00] T. Hughes, M. Marton, A. Jones, C. Roberts, R. Stoughton, C. Armour, H. Bennett, E. Coffey, H. Dai, and Y. He. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, 2000.

[HSK98a] K. R. Heidtke and S. Schulze-Kremer. BioSim–a new qualitative simulation environment for molecular biology. *Proc Int Conf Intell Syst Mol Biol*, 6:85–94, 1998.

[HSK98b] K. R. Heidtke and S. Schulze-Kremer. Design and implementation of a qualitative simulation model of lambda phage infection. *Bioinformatics*, 14:81–91, 1998.

[HSWK02] Stephen E. Harris, Bruce K. Sawhill, Andrew Wuensche, and Stuart Kauffman. A model of transcriptional regulatory networks based on biases in the observed regulation rules. *Complex.*, 7(4):23–40, 2002.

[HTPDlR07] J. Hernandez-Toro, C. Prieto, and J. De las Rivas. APID2NET: unified interactome graphic analyzer. *Bioinformatics*, 23:2495–2497, Sep 2007.

[ICF+90] S Iuchi, V Chepuri, H A Fu, R B Gennis, and E C Lin. Requirement for terminal cytochromes in generation of the aerobic signal for the arc regulatory system in Escherichia coli: study utilizing deletions

and lac fusions of cyo and cyd. *J Bacteriol*, 172(10):6020–6025, Oct 1990.

[JGC+05] PE Jacques, AL Gervais, M Cantin, JF Lucier, G Dallaire, G Drouin, L Gaudreau, J Goulet, and R Brzezinski. MtbRegList, a database dedicated to the analysis of transcriptional regulation in mycobacterium tuberculosis. *Bioinformatics*, 21(10):2563–5, 2005.

[JI98] M Jishage and A Ishihama. A stationary phase protein in Escherichia coli with binding activity to the major sigma subunit of RNA polymerase. *Proc Natl Acad Sci U S A*, 95(9):4953–4958, Apr 1998.

[Kau93] SA Kauffman. *The origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.

[KCVGC+05] I. M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I. T. Paulsen, M. Peralta-Gil, and P. D. Karp. EcoCyc: a comprehensive database resource for Escherichia coli. *Nucleic Acids Res.*, 33:D334–337, Jan 2005.

[Kin04] Robert Kincaid. VistaClara: an interactive visualization for exploratory analysis of DNA microarrays. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 167–174, New York, NY, USA, 2004. ACM.

[Kit02] H Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, Mar 2002.

[KPST69] S A Kauffman, C Peterson, B Samuelsson, and C Troein. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467, 1969.

[KPST03] Stuart Kauffman, Carsten Peterson, Bjorn Samuelsson, and Carl Troein. Random Boolean network models and the yeast transcriptional network. *Proc Natl Acad Sci U S A*, 100(25):14796–9, 2003.

[Kui86] B Kuipers. Qualitative simulation. *"Artif. Intell."*, 29:289–388, 1986.

[Kui94] B Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge, MA, 1994.

[LBP07] M Le Borgne and V Philippe. Decision diagrams for qualitative biological models. Technical Report RR-6182, INRIA, 2007.

[Lec08] R. D. Leclerc. Survival of the sparsest: robust gene networks are parsimonious. *Mol. Syst. Biol.*, 4:213, 2008.

[LFGL99] RJ Lipshutz, SP Fodor, TR Gingeras, and DJ Lockhart. High density synthetic oligonucleotide arrays. *Nat Genet*, 21:20–4, 1999.

[LRR+02] Tong Ihn Lee, Nicola J Rinaldi, Francois Robert, Duncan T Odom, Ziv Bar-Joseph, Georg K Gerber, Nancy M Hannett, Christopher T Harbison, Craig M Thompson, Itamar Simon, Julia Zeitlinger, Ezra G Jennings, Heather L Murray, D Benjamin Gordon, Bing Ren, John J Wyrick, Jean-Bosco Tagne, Thomas L Volkert, Ernest Fraenkel, David K Gifford, and Richard A Young. Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, 298(5594):799–804, 2002.

[LW00] D. J. Lockhart and E. A. Winzeler. Genomics, gene expression and DNA arrays. *Nature*, 405:827–836, Jun 2000.

[MACV03] A. Martínez-Antonio and J. Collado-Vides. Identifying global regulators in transcriptional regulatory networks in bacteria. *Curr. Opin. Microbiol.*, 6:482–489, Oct 2003.

[MAJSCV06] A. Martínez-Antonio, S. C. Janga, H. Salgado, and J. Collado-Vides. Internal-sensing machinery directs the activity of the regulatory network in Escherichia coli. *Trends Microbiol.*, 14:22–27, Jan 2006.

[MBLBLG00] H. Marchand, P. Bournai, M. Le Borgne, and P. Le Guernic. Synthesis of discrete-event controllers based on the signal environment. *Discrete Event Dynamic System: Theory and Applications*, 10(4):325–246, 2000.

[MDH+04] Sanie Mnaimneh, Armaity P Davierwala, Jennifer Haynes, Jason Moffat, Wen-Tao Peng, Wen Zhang, Xueqi Yang, Jeff Pootoolal, Gordon Chua, Andres Lopez, Miles Trochesset, Darcy Morse, Nevan J Krogan, Shawna L Hiley, Zhijian Li, Quaid Morris, Jorg Grigull, Nicholas Mitsakakis, Christopher J Roberts, Jack F Greenblatt, Charles Boone, Chris A Kaiser, Brenda J Andrews, and Timothy R Hughes. Exploration of essential gene functions via titratable promoter alleles. *Cell*, 118(1):31–44, 2004.

[MHB+03] R Münch, K Hiller, H Barg, D Heldt, S Linz, E Wingender, and D Jahn. PRODORIC: prokaryotic database of gene regulation. *Nucleic Acids Res.*, 31(1):266–9, 2003.

[MKD+04] H. W. Ma, B. Kumar, U. Ditges, F. Gunzer, J. Buer, and A. P. Zeng. An extended transcriptional regulatory network of Escherichia coli and analysis of its hierarchical structure and network motifs. *Nucleic Acids Res.*, 32:6643–6649, 2004.

[MMT+08] P. T. Monteiro, N. D. Mendes, M. C. Teixeira, S. d'Orey, S. Tenreiro, N. P. Mira, H. Pais, A. P. Francisco, A. M. Carvalho, A. B. Lourenço, I. Sá-Correia, A. L. Oliveira, and A. T. Freitas. YEASTRACT-DISCOVERER: new tools to improve the analysis of transcriptional

regulatory associations in Saccharomyces cerevisiae. *Nucleic Acids Res.*, 36:D132–136, Jan 2008.

[MNB+06] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7 Suppl 1(NIL):S7, 2006.

[MS00] G. MacBeath and S. L. Schreiber. Printing proteins as microarrays for high-throughput function determination. *Science*, 289:1760–1763, Sep 2000.

[MTAB99] L. Mendoza, D. Thieffry, and E. R. Alvarez-Buylla. Genetic control of flower morphogenesis in Arabidopsis thaliana: a logical analysis. *Bioinformatics*, 15:593–606, 1999.

[MWG+06] Kenzie D MacIsaac, Ting Wang, D Benjamin Gordon, David K Gifford, Gary D Stormo, and Ernest Fraenkel. An improved map of conserved regulatory sites for Saccharomyces cerevisiae. *BMC Bioinformatics*, 7(NIL):113, 2006.

[NBA+07] H Neuweger, J Baumbach, S Albaum, T Bekel, M Dondrup, AT Hüser, J Kalinowski, S Oehm, A Pühler, S Rahmann, J Weile, and A Goesmann. Corynecenter - an online resource for the integrated analysis of corynebacterial genome and transcriptome data. *BMC Syst Biol.*, 1:55, 2007.

[NGBBK02] Nabil Nabil Guelzim, Samuele Bottani, Paul Bourgine, and François 2 Képès. Topological and causal structure of the yeast transcriptional regulatory network. *Nature Genetics*, 31:60–63, 2002.

[NIE+04] D. E. Nelson, A. E. Ihekwaba, M. Elliott, J. R. Johnson, C. A. Gibney, B. E. Foreman, G. Nelson, V. See, C. A. Horton, D. G. Spiller, S. W. Edwards, H. P. McDowell, J. F. Unitt, E. Sullivan, R. Grimley, N. Benson, D. Broomhead, D. B. Kell, and M. R. White. Oscillations in NF-kappaB signaling control the dynamics of gene expression. *Science*, 306:704–708, Oct 2004.

[NTIM05] Naoki Nariai, Yoshinori Tamada, Seiya Imoto, and Satoru Miyano. Estimating gene regulatory networks and protein-protein interactions of Saccharomyces cerevisiae from multiple genome-wide data. *Bioinformatics*, 21 Suppl 2(NIL):ii206–ii212, 2005.

[ODB00] N Ogawa, J DeRisi, and P O Brown. New components of a system for phosphate accumulation and polyphosphate metabolism in Saccharomyces cerevisiae revealed by genomic expression analysis. *Mol Biol Cell*, 11(12):4309–21, 2000.

[OS02] AJ Obaya and JM Sedivy. Regulation of cyclin-cdk activity in mammalian cells. *Cell Mol Life Sci.*, 59(1):126–42, 2002.

[PRKG+99] I Perez-Roger, SH Kim, B Griffiths, A Sewing, and H Land. Cyclins D1 and D2 mediate myc-induced proliferation via sequestration of p27(Kip1) and p21(Cip1). *EMBO J.*, 18(19):5310–20, 1999.

[PS92] CO Pabo and RT Sauer. Transcription factors: structural families and principles of DNA recognition. *Annu Rev*, 61:1053–95, 1992.

[RGKS03] T. Rohwerder, T. Gehrke, K. Kinzler, and W. Sand. Bioleaching review part A: progress in bioleaching: fundamentals and mechanisms of bacterial metal sulfide oxidation. *Appl. Microbiol. Biotechnol.*, 63:239–248, Dec 2003.

[RH06] George G Roberts and Alan P Hudson. Transcriptome profiling of Saccharomyces cerevisiae during a transition from fermentative to glycerol-based respiratory growth reveals extensive metabolic and structural remodeling. *Mol Genet Genomics*, 276(2):170–86, 2006.

[RLS+06] Ovidiu Radulescu, Sandrine Lagarrigue, Anne Siegel, Philippe Veber, and Michel Le Borgne. Topology and static response of interaction networks in molecular biology. *J R Soc Interface*, 3(6):185–96, 2006.

[RRAS08] L. Royer, M. Reimann, B. Andreopoulos, and M. Schroeder. Unraveling protein networks with power graph analysis. *PLoS Comput. Biol.*, 4:e1000108, 2008.

[RZD+07] F Rapaport, A Zinovyev, M Dutreix, E Barillot, and JP Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8:35, 2007.

[RZL07] O Radulescu, A Zinovyev, and A Lilienbaum. Model reduction and model comparison for nfkb signaling. In *Foundations of Systems Biology in Engineering, FOSBE'07*, Stuttgart, Germany, 2007.

[SD07] M. Singhal and K. Domico. CABIN: collective analysis of biological interaction networks. *Comput Biol Chem*, 31:222–225, Jun 2007.

[SE05] Bulashevskan S. and R. Eils. Inferring genetic regulatory logic from expression data. *Bioinformatics*, 21(11):2706–13, 2005.

[SEJGM02] B. Schoeberl, C. Eichler-Jonsson, E. D. Gilles, and G. Müller. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nat. Biotechnol.*, 20:370–375, Apr 2002.

[SGCPG*al.*06] Heladia Salgado, Socorro Gama-Castro, Martin Peralta-Gil, and *al.* RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Res*, 34(Database issue):D394–7, 2006.

[SIBW00] P Sudarsanam, V R Iyer, P O Brown, and F Winston. Whole-genome expression analysis of snf/swi mutants of Saccharomyces cerevisiae. *Proc Natl Acad Sci U S A*, 97(7):3364–9, 2000.

[SMdHN08] N Sierro, Y Makita, MJL de Hoon, and K Nakai. DBTBS: a database of transcriptional regulation in Bacillus subtilis containing upstream intergenic conservation information. *Nucleic Acids Res.*, 36:D93–D96, 2008.

[SMO+03] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 2003. Availability: http://www.cytoscape.org.

[SNS02] P. Simons, I. Niemelä, and T. Soininen. Extending and implementing the stable model semantics. *Artificial Intelligence*, 138(1-2):181–234, 2002.

[SOMMA02] Shai S Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. Network motifs in the transcriptional regulation network of Escherichia coli. *Nat Genet*, 31(1):64–8, 2002.

[SRB+06] A Siegel, O Radulescu, M Le Borgne, P Veber, J Ouy, and S Lagarrigue. Qualitative analysis of the relation between DNA microarray data and behavioral models of regulation networks. *Biosystems*, 84(2):153–74, 2006.

[SRN07] J. A. Sepulchre, S. Reverchon, and W. Nasser. Modeling the onset of virulence in a pectinolytic bacterium. *J. Theor. Biol.*, 244:239–257, Jan 2007.

[SSR+03] Eran Segal, Michael Shapira, Aviv Regev, Dana Pe'er, David Botstein, Daphne Koller, and Nir Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet*, 34(2):166–76, 2003.

[ST01] L. Sánchez and D. Thieffry. A logical analysis of the Drosophila gap-gene system. *J. Theor. Biol.*, 211:115–141, Jul 2001.

[Str94] SH Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering.* Perseus Books, 1994.

[Syr] T. Syrjänen. Lparse 1.0 user's manual. http://www.tcs.hut.fi/Software/smodels/lparse.ps.gz.

[SZN+08] G Stoll, A Zinovyev, E Novikov, L Martignetti, E Barillot, F Tirode, K Laud, N Guillon, D Surdez, O Delattre, O Radulescu, T Baumuratova, S. Blanchon, M Le Borgne, P Veber, C Guziolowski, and A Siegel. Systems biology sitcon project for studying ewing sarcoma. In *International Conference on Systems Biology*, Goteborg, Sweden, 2008.

[TB04] SA Teichmann and MM Babu. Gene regulatory network growth by duplication. *Nat Genet*, 36:492–6, 2004.

[TCN01] J. J. Tyson, K. Chen, and B. Novak. Network dynamics and cell physiology. *Nat. Rev. Mol. Cell Biol.*, 2:908–916, Dec 2001.

[Tho91] R Thomas. Regulatory networks seen as asynchronous automata : a logical description. *J. Theor. Biol.*, 153:1–23, 1991.

[TMD03] L Travé-Massuyès and P Dague. Modèles et raisonnements qualitatifs. page 364. Hermes Sciences, Paris, 2003.

[TMJ+06] M. C. Teixeira, P. Monteiro, P. Jain, S. Tenreiro, A. R. Fernandes, N. P. Mira, M. Alenquer, A. T. Freitas, A. L. Oliveira, and I. Sá-Correia. The YEASTRACT database: a tool for the analysis of transcription regulatory associations in Saccharomyces cerevisiae. *Nucleic Acids Res.*, 34:D446–451, Jan 2006.

[TTK95] R. Thomas, D. Thieffry, and M. Kaufman. Dynamical behaviour of biological regulatory networks–I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bull. Math. Biol.*, 57:247–276, Mar 1995.

[VBSR05] Philippe Veber, Michel Le Borgne, Anne Siegel, and Ovidiu Radulescu. Complex qualitative models in biology: A new approach. *Complexus*, 2:3–4, 2004/2005.

[Veb07] P Veber. *Modélisation grande échelle de réseaux biologiques : vérification par constraintes booléenes de la cohérence des donées.* PhD thesis, Université de Rennes 1, 2007.

[VGLB+08] P Veber, C Guziolowski, M Le Borgne, O Radulescu, and A Siegel. Inferring the role of transcription factors in regulatory networks. *BMC Bioinformatics*, 9:228, 2008.

[VP94] A. Varma and B. O. Palsson. Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. *Bio/Technology*, 12:994–998, 1994.

[VZVK95] VE Velculescu, L Zhang, B Vogelstein, and KW Kinzler. Serial analysis of gene expression. *Science*, 270:467, 1995.

[XvdL05] Biao Xing and Mark J van der Laan. A causal inference approach for constructing transcriptional regulatory networks. *Bioinformatics*, 21(21):4007–13, 2005.

[YIJ04] Chen-Hsiang Yeang, Trey Ideker, and Tommi Jaakkola. Physical network models. *J Comput Biol*, 11(2-3):243–62, 2004.

[YMM+05] Chen-Hsiang Yeang, H Craig Mak, Scott McCuine, Christopher Workman, Tommi Jaakkola, and Trey Ideker. Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biol*, 6(7):R62, 2005.

[YWHT08] L. Yang, J. R. Walker, J. B. Hogenesch, and R. S. Thomas. NetAtlas: a Cytoscape plugin to examine signaling networks based on tissue gene expression. *In Silico Biol. (Gedrukt)*, 8:47–52, 2008.

[YY71] G. Yagil and E. Yagil. On the relation between effector concentration and the rate of induced enzyme synthesis. *Biophys. J.*, 11:11–27, Jan 1971.

Scientific activities

6.3 Publications

Peer reviewed journal articles

- Guziolowski, C., Bourdé, A., Moreews, F., Siegel, A., "BioQuali Cytoscape Plugin: Analysing the global consistency of regulatory networks", *BMC Genomics*, *BMC Genomics*, vol. 10, pp. 244, 2009.

- Veber, P., Guziolowski, C., Le Borgne, M., Radulescu, O., Siegel, A., "Inferring the role of transcription factors in regulatory networks", *BMC Bioinformatics*, vol. 9, pp. 228, 2008.

- Guziolowski, C., Veber, P., Le Borgne, M., Radulescu, O., Siegel, A., "Checking Consistency Between Expression Data and Large Scale Regulatory Networks: A Case Study", *Journal of Biological Physics and Chemistry*, vol. 7, pp. 37-43, 2007.

- Didier, G. and Guziolowski, C., "Mapping Sequences by Parts", *Algorithms for Molecular Biology*, vol. 2, pp. 11, 2007.

- Latorre, M., Silva, H., Saba, J., Guziolowski, C., Vizoso, P., Martinez, V., Maldonado, J., Morales, A., Caroca, R., Cambiazo, V., Campos-Vargas, R., Gonzalez, M., Orellana, A., Retamales, J., Meisel, L.A, "JUICE: a data management system that facilitates the analysis of large volumes of information in an EST project workflow", *BMC Bioinformatics*, vol. 7, pp. 513, 2006.

Conference proceedings

- Blachon, S., Stoll, G., Guziolowski, C., Zinovyev, A., Barillot, E., Siegel, A., Radulescu, O., "Method for relating inter-patient gene copy numbers variations with gene expression via gene influence network" *BMIINT: Biomedical Informatics and Intelligent Methods in the Support of Genomic Medicine.* Thessaloniki, Greece, April 2009, AIAI, pp 72-87.

- Guziolowski, C., Gruel, J., Radulescu, O., Siegel, A. "Curating a large-scale regulatory network by evaluating its consistency with expression datasets", *CIBB'08: 5th International Conference on Bioinformatics and Biostatistics*, Salerno, Italy

2008, Lecture Notes in Computer Science, vol. 5488, pp. 144-155, Springer (selected paper).

- Siegel, A., Guziolowski, C., Veber, P., Radulescu, O., Le Borgne, M., "Qualitative response of interaction networks: application to the validation of biological models", Contribution to minisymposium New Research in Bioinformatics. ICIAM'07: 6th International Congress on Industrial and Applied Mathematics, Zurich 2007, PAMM, vol. 7, no. 1, pp. 1121803-1121804.

Submitted

- Guziolowski, C., Blachon, S., Radulescu, O., Baumuratova, T., Stoll, G., Siegel, A., "Designing logical rules to model the response of biomolecular networks with complex interactions: an application to cancer modeling", *IEEE/ACM Transactions on Computational Biology and Bioinformatics.*

- Gebser, M., Guziolowski, C., Schaub, T., Siegel, A., Thiele, T., Veber, P., "Prediction and Repair in Large Biological Networks with Answer Set Programming", *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning.* Toronto, Canada, May 9-13, 2010.

Non peer reviewed publications

- Guziolowski, C., Veber, P., Le Borgne, M., Radulescu, O., Siegel, A. "Analysing regulatory networks using a qualitative approach", *Genomes to Systems*, Manchester, UK, 2008 (poster session).

- Siegel, A., Guziolowski, C., Veber, P., Radulescu, O., Le Borgne, M., "Optimiser un plan d'expérience à partir de modèles qualitatifs?", *BioFutur* vol. 275, pp. 27, 2006

Oral communications

- Workshop in Gen2Bio: Les rencontres Biotech organisées par OUEST-genopole: "Plugin BioQuali pour Cytoscape: un nouvel outil mis en place sur la plate-forme" (La Boule, France).

- JOBIM 2008 satellite meeting: Dynamical modelling and simulation of biological networks: "Adding missing post-transcriptional regulations to a regulatory network" (Lille, France).

- Seminaire Bioinformatique at Symbiose Team, INRIA - Centre Bretagne Atlantique: "Mapping sequences by parts" (Rennes, France).

- Cinquièmes Rencontres autour de la plate-forme Bio-informatique Genouest: "BioQuali: tool for analysing regulatory networks" (Rennes, France).

- 2nd International Course in Yeast Systems Biology, ICYSB: "Inferring the role of transcription factors in regulatory networks", (Goteborg, Sweden).
- RIAMS'06: Réseaux d'interaction, analyse, modélisation et simulation "Checking Consistency Between Expression Data and Large Scale Regulatory Networks: A Case Study" (Lyon, France).

Research in-team

- "Qualitative functions to better model the EWS/FLI-1 signalling network", SIT-CON project meeting, Institut Curie, Paris, December, 2008.
- "Adding missing post-transcriptional regulations to a regulatory network", SIT-CON project meeting, Institut Curie, Paris, May, 2008.

6.4 Academic visits

- Centre of Mathematical Modelling - Universidad de Chile, Santiago, Chile. 4 weeks visit (2009).
- Potsdam University, Potsdam, Germany. 2 weeks visit (2009).

6.5 Teaching

- Exercise and practical sessions on UML and Java programming, Ecole Nationale de la Statistique et de l'Analyse de l'Information, Bruz, France, 90 hours (2007-2009)
- Exercise and practical sessions on Introduction to Algorithmic and Programming Languages, Universidad de Chile, Santiago, Chile, annual course (2002).

List of Figures

List of Algorithms

Résumé

Il existe plusieurs approches qui modélisent des réseaux de régulation génétiques afin d'élucider la dynamique d'un système biologique. Cependant, ces approches concernent des modèles à petite-échelle. Dans cette thèse nous utilisons un approche formelle sur les réseaux de régulation à grande-échelle qui modélise les variations des concentrations des molécules d'une cellule entre deux états stationnaires. On teste la cohérence entre la topologie du réseau et des données d'expression génétique en utilisant une règle causale de consistance. Les résultats de cette approche sont : test de la consistance entre les données et un réseau, diagnostic des régions du réseau inconsistantes avec les données expérimentales, et inférence des variations des éléments du réseau. Notre méthode raisonne sur la topologie globale du réseau en utilisant des algorithmes efficaces basés sur des diagrammes de décision, des graphes de dépendance, ou la programmation par ensemble réponse. Nous avons proposé des programmes et des outils bioinformatiques basés sur ces algorithmes qui automatisent ces raisonnements. On a validé cette approche en utilisant des réseau transcriptionels des espèces *E. coli* et *S. cerevisiae*, et le réseau de signalisation de l'oncogène EWS-FLI1. Nos résultats principaux sont: (1) un pourcentage élevé de validation des prédiction sur la variation des molécules du réseau, (2) des corrections manuelles et automatiques efficaces du modèle et/ou données, (3) l'inférence automatique des rôles des facteurs de transcription, et (4) raisonnement automatique sur les causes qui influencent des phénotypes importants dans des réseaux de signalisation.

Abstract

To this date many approaches exist that model a genetic regulatory network in order to elucidate the dynamics of the system. These methods focus, however, mainly on small-scale regulatory models. In this thesis we use a formal approach over qualitative large-scale regulatory networks, that models the equilibrium shift of the cell molecules between two steady states. We test the coherency between the network topology and gene expression data, by using a general interaction logical causal rule. The outputs of our approach are to measure the consistency of our data, diagnose inconsistent regions of the network with respect to the experimental data, and infer the qualitative variation of new network molecules. Our method reasons over the whole network of interactions using efficient algorithms based either on decision diagrams, dependency graphs, or answer set programming. We proposed programs and bioinformatic tools that, based on these efficient implementations, automatize this reasoning. We validated this approach using the transcriptional networks of *E. coli* and *S. cerevisiae*, and the signaling network of the EWS-FLI1 human oncogene. Our main results were: (1) high prediction accuracy of the shifts of the network molecules, (2) effective manual and automatic corrections of the model and/or data, (3) automatic inference of the role of transcription factors, and (4) automatic reasoning over the causes that influence important phenotypes on a signaling network. All in all, we provided a methodology that can be applied to complete regulatory networks built at different molecular levels, by exploiting the constantly increasing high-throughput outputs.