Nouvelles approches informatiques et mathématiques pour la résolution de problèmes biologiques

Mohamed Lemine Ahmed Sidi

UNIVERSITÉ DE TOURS

ÉCOLE DOCTORALE MIPTIS

Laboratoire d'Informatique Fondamentale et Appliquée de Tours (EA 6300)

UNIVERSITÉ DE NOUAKCHOTT

Unité de Recherche Calcul Scientifique, Informatique et Data Science de l'Université de Nouakchott

THÈSE présentée par :

Mohamed Lemine AHMED SIDI

soutenue le 17 juin 2022
pour obtenir le grade de Docteur de l'Université de Tours et de l'Université de Nouakchott

Discipline/Spécialité : Informatique

Nouvelles approches informatiques et mathématiques pour la résolution de problèmes biologiques

Thèse dirigée par :

Emmanuel Néron	Professeur, Université de Tours
Mohamedade Farouk Nanne	Professeur, Université de Nouakchott

Rapporteurs :

Clarisse Dhaenens	Professeur, Université de Lille
Simon de Givry	Chargé de recherche INRAE, HdR, Unité de Mathématiques et Informatique Appliquées de Toulouse

Jury :

Clarisse Dhaenens	Professeur, Université de Lille
Simon de Givry	Chargé de recherche INRAE, HdR, Unité de Mathématiques et Informatique Appliquées de Toulouse
Eric Bourreau	Maître de conférences, Université de Montpellier
Dritan Nace	Professeur, Université de Technologie de Compiègne
Ronan Bocquillon	Maître de conférences, Université de Tours
Emmanuel Néron	Professeur, Université de Tours
Mohamedade Farouk Nanne	Professeur, Université de Nouakchott
Ahmedou Haouba	Professeur, Université de Nouakchott (Président du jury)

Remerciements

Je remercie chaleureusement toutes les personnes qui m'ont aidé pendant l'élaboration de ma thèse et notamment mes encadrants : Ronan BOCQUILLON, Hafedh MOHAMED BABOU, Cheikh DHIB, Emmanuel NÉRON, Ameur SOUKHAL et Mohamedade Farouk NANNE. Malgré mon envie d'avoir tous mes encadrants dans la page de garde du manuscrit, je suis fier d'être co-auteur avec ces grands chercheurs dans des publications scientifiques. Je salut leur disponibilité, leur rigueur, leur intelligence et leur patience.

Merci aux rapporteurs, Clarisse DHAENENS et Simon DE GIVRY d'avoir accepté de lire et évaluer ce manuscrit.

Merci à Ahmed HAOUBA, Dritan NACE et Eric BOURREAU qui m'ont honoré par leur participation au jury de ma thèse.

Merci au Service de Coopération et d'Action Culturelle de l'ambassade de France en Mauritanie, qui m'a permis, grâce à une allocation doctorale (18 mois pendant 3 ans), de me consacrer sereinement à l'élaboration de ma thèse.

Merci à Alfa Conseils : l'entreprise dans la quelle j'ai fait mon PFE de Master et qui était mon laboratoire de recherche à Nouakchott pendant mes années de thèse. Un merci en particulier à Oumar BELLAL, Directeur de Alfa.

Je tiens à adresser mes sincères remerciements à tous les permanents de l'équipe ROOT. Un merci en particulier à Jean-Charles BILLAUT, Tifenn RAULT, Vincent T'KINDT et Yannick KERGOSIEN.

Je remercie aussi tous les permanents et les personnels du LIFAT, de PolytechTours et de l'école doctorale. Un merci en particulier à Jean-Yves RAMEL, Christelle GRANGE, Annie SIMON, Cecile BOYER, Sébastien BEAUFILS, Gerald MAYAUD, Bénédite RICHARD et Isabelle FOULON.

Je remercie mes collègues au LIFAT. Un merci en particulier à Alexis, Boukhalfa, Cuoc, David, Gaëtan, Guillaume, Hugo, Limeme, Meya, Mostapha, Olivier, Phat, Qong, Romain et Valentine.

Mes vifs remerciements s'adressent à tous mes anciens professeurs : de la Mahdara à l'Université.

Je souhaite remercier mes amis. J'exprime ma gratitude particulièrement à Abdellahi AHMED SIDI (mon neveu) et El-hacen DIALLO.

Je remercie mes parents pour leur soutien permanent. Enfin, je remercie ma famille qui ont toujours été à mes côtés pour me soutenir. Un merci en particulier à mes frères Telmidi, Ahmedou, Abderrahmane et ma seour Khadija.

Résumé

La biologie des systèmes est un domaine récent proposant d'étudier les organismes vivants tels qu'ils se présentent en réalité. Cette approche s'oppose aux précédentes en intégrant différents niveaux d'information (biologiques, physiologiques, biochimiques, *etc.*) pour comprendre le fonctionnement de ces organismes. Cela nécessite des algorithmes de traitement et d'analyse de plus en plus spécialisés et efficaces. De nombreuses approches destinées à la comparaison de réseaux biologiques (homogènes ou hétérogènes) reposent sur des modèles de graphe. En effet, en bio-informatique, ces réseaux sont souvent modélisés par des graphes. Les sommets sont des composants biologiques et les arêtes représentent leurs interactions. En fonction de la nature de ces réseaux, les graphes correspondants peuvent être orientés ou non-orientés. Cette thèse en Recherche Opérationnelle s'inscrit dans le cadre applicatif de la biologie des systèmes et porte plus particulièrement sur un problème relatif aux réseaux biologiques hétérogènes. L'objectif principal est d'étudier la relation entre le métabolisme et le génome. Nous avons choisi de nous concentrer sur la détection de réactions métaboliques voisines catalysées par des produits de gènes voisins, où la notion de voisinage peut être modulée en autorisant que certaines réactions et/ou gènes soient omis. Dans un premier temps, nous proposons des approches permettant de comparer deux réseaux biologiques modélisés respectivement par un graphe orienté D et un graphe non-orienté G, construit sur le même ensemble de sommets. Ces approches consistent à identifier une structure biologiquement significative dans D, dont les sommets induisent dans G une structure qui soit aussi biologiquement significative. Dans un second temps, nous étudions le degré de conservation des motifs métaboliques et génomiques ainsi extraits à l'échelle de plusieurs espèces.

Mots-clefs : recherche opérationnelle, bio-informatique, comparaison de réseaux biologiques, procédure par séparation et évaluation, programmation linéaire en nombres entiers, programmation par contraintes.

Abstract

Systems biology is a recent field of science that proposes the study of living organisms as they are found in nature. This approach differs from previous ones by integrating different levels of information (biological, physiological, biochemical, etc.) to understand the functions of these organisms. This requires the use of specialized and efficient treatment and analysis algorithms. Many approaches for the comparison of biological networks (homogeneous or heterogeneous) are based on graph models. In bioinformatics, these networks are often modeled by graphs. The vertices are biological components and the edges represent their interactions. Depending on the nature of these networks, the corresponding graphs can be oriented or non-oriented. This thesis in Operational Research is in the context of systems biology and focuses on a problem related to heterogeneous biological networks. The main objective is to study the relationship between metabolism and genome. We have chosen to focus on the detection of neighboring reactions catalyzed by products of neighboring genes, where the notion of neighborhood can be modulated by allowing some reactions and/or genes to be skipped. The process comprises of two steps. Firstly, we propose approaches to compare two biological networks modeled respectively by a directed graph D and a non-oriented graph G, built on the same set of vertices. These approaches consist in identifying a biologically significant structure in D, whose vertices induce in G a structure that is also biologically significant. secondly, we study the degree of conservation of metabolic and genomic motifs at the scale of several species.

Keywords : operational research, bioinformatics, comparison of biological networks, branch-and-bound, integer linear programming, constraints programming.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

À fin de comprendre le fonctionnement du monde vivant, les scientifiques, au long de l'histoire, l'ont étudié de différentes manières, en particulier, dans l'étude des organismes bactériens qui sont parmi les plus simples en biologie, mais qui restent malgré tout étonnement complexes. Les scientifiques analysent sous plusieurs angles ce qui se passe à l'intérieur d'une cellule. De nombreux aspects de la cellule ont été scrutés, chacun correspondant à un domaine de la science de la vie (*e.g.* protéomique, génétique, génomique, transcriptomique, métabolomique, ...). Par exemple, l'aspect protéomique étudie les fonctions des protéines. Une approche simple pour obtenir de l'information à partir de séquences inconnues d'une protéine est de comparer ces séquences avec celles déjà connues, avec l'hypothèse de la corrélation entre similarité des séquences et similarité des fonctions de la protéine. Dans ce contexte, l'étude de la fonction de chaque composant biologique par la comparaison de séquences représente un des domaines les plus actifs de la bio-informatique actuelle [Myers, 1991, Waterman, 1998, Loweth, 1997, Gusfield, 1997]. Avec les nouvelles technologies, l'acquisition de données biologiques (*i.e.* métaboliques, génomiques, ...) a pu progresser de manière significative et ainsi suite au projet génome humain [Collins et al., 1998] de nombreux laboratoires ont maintenant la capacité de séquencer plus de 100 millions de paires de bases par an avec un coût réduit. La Figure 1.1 représente le coût du séquençage d'un génome en fonction du temps, passant de plus de 95 millions de dollars en juillet 2001 à moins de 1100 dollars en janvier 2020. Dans ce contexte, on assiste à une accumulation de données accessibles alors que les outils d'analyse et d'extraction d'informations pertinentes font souvent défaut. L'explosion des données a nécessité le développement d'approches plus systémiques de l'analyse des données.

Les approches classiques se focalisent sur l'analyse des données biologiques en étudiant de manière indépendante les différents aspects (protéomiques, génomiques, génétiques, ...). Or, la compréhension d'un système vivant dans son ensemble nécessite la prise en compte de ces différents aspects simultanément. La biologie systémique se charge de cette mission. Elle vise à comprendre les entités biologiques au niveau systémique, en les analysant non seulement comme des composants individuels, mais aussi comme des systèmes en interaction ayant des propriétés émergentes. On passe alors de l'étude d'un composant biologique à l'étude des relations entre les composants.

L'une des méthodes utilisées pour mieux comprendre les liens complexes entre le génotype (ensemble des gènes relevés chez un individu) et le phénotype (ensemble des caractères observés chez un individu) consiste à étudier les relations entre différents composants biologiques (entre les protéines, entre les métabolites, ...). Celles-ci forment ce qui est appelé des réseaux biologiques. Les systèmes biologiques sont souvent représentés comme des réseaux qui sont des ensembles complexes d'interactions ou de relations binaires entre différentes entités. Certains réseaux biologiques modélisent les fonctions des interactions moléculaires spécifiques aux cellules et aux tissus à un niveau d'organisation cellulaire, allant des cellules à un organe complet. Parmi les types de réseaux biologiques les plus connus sont les réseaux d'interaction protéine-protéine, réseaux métaboliques, réseaux d'interactions génétiques. Les

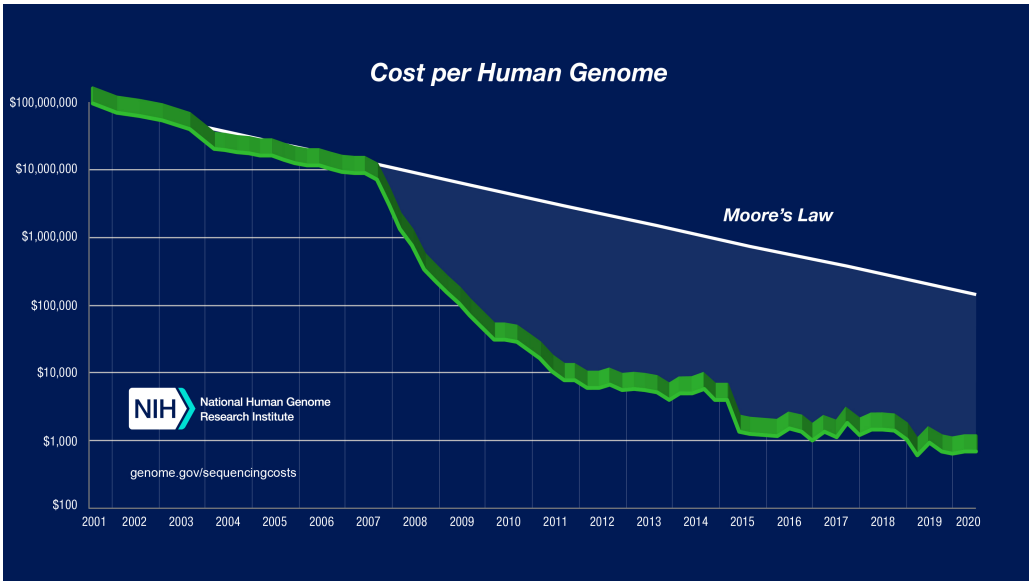


FIG. 1.1 : Évolution temporelle du prix du séquençage. Depuis 2001, le prix pour séquencer un génome n'a cessé de décroître. Entre 2007 et 2008, la diminution s'est accélérée, conduisant à la création de plus en plus de données. La loi de Moore symbolise l'évolution parallèle de la puissance de calcul disponible pour traiter ces données. On voit qu'après 2008, l'évolution des ressources informatiques ne suit plus la diminution des coûts de production génomique : l'évolution de la puissance brute informatique n'est plus suffisante pour traiter l'ensemble des données générées ; de nouveaux algorithmes doivent être mis au point.
Source : www.genome.gov/sequencingCosts

réseaux principalement utilisés dans cette thèse sont les réseaux liant les protéines, le génome ainsi que les réseaux métaboliques.

Du point de vue modélisation, nous représentons un réseau biologique par un graphe, où les sommets représentent les composants biologiques, et où les arêtes représentent les interactions entre les composants correspondants. La diversité de réseaux biologiques et de leurs modélisations a donné lieu à deux types de comparaison. La première consiste à comparer des réseaux de même type (par exemple, réseaux d'interaction protéine-protéine de deux espèces). Le second type consiste à comparer le même aspect représenté de façons différentes pour une même espèce.

Un problème de la comparaison de réseaux homogènes largement étudié dans la littérature consiste à déterminer si une certaine requête, représentée par un chemin, un arbre ou un graphe, est présente dans le graphe représentant le réseau biologique. Il permet en effet d'identifier les parties importantes d'un grand réseau ou encore de découvrir des fonctions communes entre différentes espèces. Cependant, ce type de problème est rapidement difficile d'un point de vue algorithmique puisqu'il est équivalent au problème d'isomorphisme de graphe [Bramwell et al., 1988]. Ce dernier conduit généralement a des problèmes d'alignement de graphes qui ont été beaucoup étudiés dans la littérature. Le principal objectif de l'alignement est de prédire la meilleure correspondance entre les réseaux. Ce type de comparaison peut être appliqués sur deux ou plusieurs réseaux [Bramwell et al., 1988, Conte et al., 2004, Kelley et al., 2003, Kelley et al., 2004].

Cette thèse aborde le second type de comparaison des réseaux biologiques qui a été peu étudié dans la littérature [Ogata et al., 2000, Fertin et al., 2012, Babou, 2012, Zaharia et al., 2019]. Dans cette thèse, on s'intéresse à la comparaison de deux réseaux hétérogènes. On cherche un chemin dans le premier graphe dont l'ensemble de sommets induit un sous graphe connexe dans le second graphe. Un exemple d'application de ce problème est l'étude les relations entre le métabolisme et le génome. Nous travaillons simultanément sur les aspects métaboliques, protéomiques et génomiques. Dans un premier temps,

nous identifions les motifs métaboliques et génomiques pour une espèce en cherchant une chaîne de réactions dans le réseau métabolique qui sont catalysées par le produit de gènes voisins. Ensuite, nous étudions le degré de conservation des motifs métaboliques et génomiques à l'échelle de plusieurs espèces.

Ce manuscrit est composé de plusieurs chapitres. Le chapitre 2 est un chapitre introductif dont l'objectif est de présenter des notions de biologie qui sont nécessaires à la compréhension de ce manuscrit et l'état de l'art sur les méthodes développées pour la comparaison de réseaux biologiques hétérogènes.

Les chapitres 3 et 4 de cette thèse portent sur une étude algorithmique des problèmes d'optimisation combinatoire issus de la comparaison des réseaux biologiques hétérogènes appelés SkewGraM. Nous proposons des méthodes algorithmiques et leur implémentation pour la résolution de deux problèmes avec des expérimentations. Dans le chapitre 3, nous traitons une variante du problème SkewGraM appelée One-to-One SkewGraM. Nous améliorons un algorithme par séparation et évaluation de la littérature nous permettant de résoudre le problème plus efficacement. Nous proposons également deux formulations de programmation linéaire en nombres entiers (ILP) en adaptant et en combinant des formulations du problème du plus long chemin et du problème de l'arbre couvrant de poids minimum. Nous proposons également un modèle de programmation par contraintes pour résoudre ce problème. Dans le chapitre 4, nous traitons une variante plus générale de ce problème. Nous adaptons les algorithmes proposés dans le chapitre 3 à ce problème. Nous évaluons ces méthodes sur des données aléatoires ainsi que des données réelles.

Dans le chapitre 5, nous exploitons les résultats obtenus par les méthodes proposées dans les chapitres 3 et 4. On étudie comment de tels motifs peuvent être analysés et regroupés à fin de révéler des contextes métaboliques et génomiques conservés entre plusieurs espèces.

Enfin, nous achevons ce manuscrit par un rappel des résultats obtenus et une présentation de nos travaux en cours de réalisation.

Chapitre 2

État de l'art

Contents

2.1 Introduction

Les problèmes étudiés durant cette thèse sont motivés par des problématiques biologiques. Du point de vue algorithmique, ces problèmes sont difficiles. Dans ce chapitre, nous présentons tout d'abord les notions biologiques nécessaires à leur compréhension. Nous présentons ensuite des travaux de la littérature connexes, portant sur la comparaison de séquences et de réseaux biologiques. Le cheminement suivi nous amène pour terminer à la définition du problème SkewGraM, sur lequel cette thèse porte.

2.2 Quelques notions de biologie

L'unité de base d'un organisme est la cellule (excepté pour les virus). Elle possède des structures lui assurant une capacité de multiplication (reproduction) et une autonomie. Il existe de nombreux sous-domaines scientifiques de la biologie (dont le nom a été créé avec le suffixe "omique") qui étudient la cellule. Cette dernière contient le génome et de nombreuses molécules biologiques. Dans cette thèse, nous étudions les organismes bactériens qui sont parmi les plus simples en biologie, mais qui restent malgré tout étonnement complexes. Ces espèces sont des micro-organismes vivants constitués généralement d'une seule cellule sans noyau. Elles sont capables de se reproduire par division cellulaire. Des bactéries sont utiles et nécessaires alors que d'autres sont pathogènes. Nous présentons ci-après quelques composants biologiques qui se trouvent à l'intérieur d'une cellule.

2.2.1 Métabolisme

Le métabolisme est un processus qui permet de produire l'énergie et les composants nécessaires à la croissance, à la reproduction et au maintien de la santé [Christakos et al., 2010, Christakos et al., 2010]. Il permet également de se débarrasser des substances toxiques. Le mot métabolisme provient du grec metabolê qui signifie changement. Le métabolisme est l'ensemble des transformations biochimiques qui se déroulent au sein de la cellule pour assurer ses besoins fonctionnels. Il est constitué de deux mécanismes opposés : le *catabolisme* qui permet d'extraire l'énergie des nutriments, par dégradation des molécules énergétiques, et l'*anabolisme* qui permet la synthèse d'éléments organiques grâce à cet apport d'énergie. Par exemple, une description de métabolisme de lipides peut être trouvée dans l'article de cette référence [Dawes and Ribbons, 1964]. Le terme métabolisme englobe plusieurs acteurs qui sont en relation avec la transformation chimique des molécules biochimiques. Réactions, métabolites, enzymes et cofacteurs sont parmi les acteurs principales dans le métabolisme.

2.2.1.1 Réactions et métabolites

Une réaction biochimique produit un groupe d'un ou plusieurs métabolites (appelés produits) à partir d'un autre groupe d'un ou plusieurs métabolites (appelés substrats) [Samal et al., 2006]. Les métabolites sont des petites molécules intervenant dans le métabolisme. Ils sont importés/exportés et/ou synthétisés/dégradés à l'intérieur d'un organisme. Les réactions chimiques peuvent se produire dans les deux sens. Cependant, dans certaines conditions physiologiques, les réactions se produisent dans une seule direction. Dans ce cas, si toutes les autres conditions restent inchangées, elles sont définies comme irréversibles. Ces réactions sont traditionnellement regroupées en voies métaboliques, qui peuvent à leur tour être classées comme anaboliques ou cataboliques. Les voies peuvent être étudiées isolément ou être combinées ensemble (puisqu'elles se chevauchent) pour donner un réseau métabolique. Les avantages de l'étude de l'ensemble du réseau plutôt que des voies individuelles sont nombreux et incluent, par exemple, la possibilité d'explorer des voies alternatives [Lacroix et al., 2008].

Dans les cellules, certaines réactions sont spontanées, mais la plupart des réactions sont catalysées par une ou plusieurs enzymes, ce qui augmente considérablement leur vitesse.

2.2.1.2 Enzymes

Les enzymes sont des protéines complexes qui contribuent à accélérer les réactions chimiques dans l'organisme vivant. Un exemple d'enzyme est l'amylase salivaire, qui hydrolyse son substrat, l'amylase, un composant de l'amidon [Bernfeld et al., 1948]. En 1908, une étude de Wohlgemuth a identifié la présence d'amylase dans l'urine, ce qui a conduit à l'utilisation de l'amylase comme test de diagnostic en laboratoire. L'amylase est un test fréquemment demandé avec la lipase, en particulier dans le cadre d'une suspicion de pancréatite aiguë [Zakowski and Bruns, 1985].

Les enzymes ont pour mission essentielle de réduire l'énergie d'activation des réactions chimiques à l'intérieur de la cellule. Pour ce faire, les enzymes se lient aux molécules réactives et les maintiennent de telle sorte que les processus de rupture et de formation des liaisons chimiques se déroulent plus facilement. Comme toutes les enzymes augmentent la vitesse de réaction, elles sont considérées comme des catalyseurs organiques. Une même réaction peut être catalysée par plusieurs enzymes, et une seule enzyme peut catalyser une ou plusieurs réactions. Les enzymes sans spécificité sont appelées enzymes promiscuous et elles peuvent accepter plusieurs substrats similaires [Nobeli et al., 2009].

Après la première identification d'une enzyme en 1833 et l'introduction du terme “enzyme" en 1876, une proposition faite par Émile Duclaux en 1898 d'ajouter le suffixe "ase" au nom du substrat pour la nomenclature d'une enzyme. Malgré cette proposition, une confusion systématique dans la dénomination des enzymes a été marquée jusqu'aux années 1950. Les enzymologistes ont commencé à s'attaquer à ce problème [Tipton and Boyce, 2000].

La seule nomenclature officielle des enzymes est celle établie par le Comité de nomenclature de l'Union internationale de Biochimie et de Biologie Moléculaire (UIBBM). Dans ce système de nomenclature, un numéro de classification appelé “EC number" est associé aux enzymes, en fonction des réactions chimiques que les enzymes catalysent [Webb, 1992]. Ce numéro se présente de la manière suivante : *EC* [numéro de la classe].[numéro de la sous-classe].[numéro de la sous-sous-classe].[numéro individuel de série dans la sous-sous classe]. Le premier chiffre représente les enzymes qui sont classées selon le mécanisme de la réaction enzymatique. Le deuxième chiffre définit la sous-classe, c'est-à-dire la nature des groupements chimiques affectés (ou les types individuels de réactions). Le troisième chiffre définit la sous-sous-classe, en pratique le substrat précis de l'enzyme ; le quatrième identifiant numérique correspond au numéro d'ordre dans la sous-sous-classe (date de caractérisation, organe de caractérisation princeps, organisme).

Exemple : Glucose oxydase : EC 1.1.3.4.
Glucose oxydase : *EC* 1.1.3.4. Ce chiffre est explicité ci-dessous :

1. *EC* 1 : Oxydoréductase
2. *EC* 1.1 : Agissant sur le groupe CH-OH du donneur
3. *EC* 1.1.3 : Avec l'oxygène comme accepteur
4. Le dernier chiffre est le numéro individuel de l'enzyme

Il est important de noter qu'un numéro *EC* n'est pas équivalent à une enzyme, ni à une réaction chimique. Les numéros *EC* décrivent simplement les réactions catalysées par les enzymes. Ce qui signifie que deux réactions distinctes peuvent avoir le même indice *EC* si elles impliquent des transformations chimiquement similaires.

Selon le type de réactions qu'elles catalysent, les enzymes sont classées en sept catégories, à savoir les oxydoréductases, les transférases, les hydrolases, les lyases, les isomérases, les ligases et les translocases. Les oxydoréductases, les transférases et les hydrolases sont les formes les plus abondantes d'enzymes *cf.* Tableau 2.1.

Numéro EC	Classe d'enzyme	Description
1.	Oxidoreductases	Catalyser la réaction d'oxydoréduction et peut être classées en oxydase et réductase.
2.	Transferases	Catalyser le transfert ou l'échange de certains groupes entre certains substrats
3.	Hydrolases	Accélérer l'hydrolyse des substrats
4.	Lyases	Favoriser l'élimination d'un groupe du substrat pour laisser une réaction de double liaison ou catalyser sa réaction inverse.
5.	Isomerases	Faciliter la conversion des isoisomères, des isomères géométriques ou des isomères optiques.
6.	Ligases	Catalyser la synthèse de deux substrats moléculaires en un composé moléculaire avec l'énergie libérée.
7.	Translocases	Catalyser le mouvement des ions ou des molécules à travers les membranes ou leur séparation à l'intérieur des membranes.

TAB. 2.1 : Classification des enzymes.

Selon le principe de classification unifiée des enzymes publié par la Société internationale de biochimie, chaque groupe d'enzymes des sept catégories ci-dessus peut être encore divisé en plusieurs sous-groupes selon les caractéristiques des groupes fonctionnels ou des liaisons dans les substrats. À fin de montrer plus précisément les propriétés des substrats ou des réactifs, chaque sous-classe est encore divisée en sous-classes et contient directement une quantité d'enzymes.

En outre, sur la base de la composition moléculaire, les enzymes peuvent être divisées en enzymes pures et en enzymes de liaison. Les enzymes contenant uniquement des protéines sont appelées enzymes pures. Les enzymes de liaison sont composées de protéines et de cofacteurs. Ce n'est que lorsque ces deux composants sont combinés que l'enzyme peut avoir une activité catalytique.

2.2.1.3 Cofacteurs

La catalyse d'une réaction par les enzymes est parfois affectée par la présence de certaines petites molécules, appelées cofacteurs. En se liant à l'enzyme, les cofacteurs peuvent augmenter ou diminuer l'activité de l'enzyme et sans le cofacteur l'enzyme reste dans les formes inactives. Une fois que le cofacteur est ajouté, l'enzyme devient actif [Kern and Zuiderweg, 2003].

2.2.1.4 L'évolution du métabolisme

Il existe quatre hypothèses pertinentes sur l'évolution du métabolisme. Parmi ces hypothèses, nous présentons le modèle rétrograde et le modèle patchwork qui sont deux grandes hypothèses qui s'affrontent pour expliquer l'évolution du métabolisme. Ces deux modèles, plutôt que de s'opposer, sont certainement complémentaires.

1. Le modèle d'évolution rétrograde a été l'une des premières théories de l'évolution des voies métaboliques, proposée par [Horowitz, 1945]. Ce modèle suppose qu'un organisme est capable d'utiliser certaines molécules de l'environnement par exemple. Les métabolites intermédiaires sont épuisés par les réactions et régénérés par de nouvelles réactions.
   Prenons l'exemple de la Figure 2.1. Le métabolite M est essentiel pour l'organisme. Il est produit à partir du substrat A, cette réaction est catalysée par l'enzyme jaune. Un organisme qui ne peut pas produire de métabolite A utilise dans un premier temps tous les métabolites A présents dans

son environnement, ce qui conduit à une forte diminution de la concentration environnementale de A et ensuite il donne à l'organisme un avantage sélectif. Cependant, comme pour A, la concentration de B dans l'environnement diminuera avec son utilisation et, par conséquent, une enzyme est sélectionnée qui catalyse la production de B à partir d'un métabolite C.
Notez que puisque l'enzyme jaune peut déjà se lier au métabolite A, il y a plus de chance que la deuxième enzyme recrutée soit le résultat d'une duplication du gène qui code pour l'enzyme jaune plutôt qu'un autre gène qui code pour une enzyme sans affinité avec A. Ce modèle est étayé par des études qui ont montré que des enzymes homologues (issus de gènes dérivés du même gène ancestral) sont significativement plus susceptibles d'être distantes de moins de 3 étapes l'une de l'autre dans le graphe métabolique que des enzymes non homologues [Alves et al., 2002].

2. Le modèle d'évolution patchwork suppose que les enzymes augmentent leur spécificité pour un substrat donné après duplication [Jensen and Meckling, 1976, Lazcano and Miller, 1996, Yang et al., 2021]. Ainsi, il est considéré initialement que les enzymes présentent une large gamme de substrats et de produits, bien que certains d'entre eux soient favorisés. Cette large gamme de substrats et produits enzymatiques permet la génération d'une large gamme de composés. La duplication de gènes dans ces voies métaboliques offre un avantage sélectif en augmentant la production du métabolite d'intérêt. Finalement, les événements de duplication sont suivis par événements de spéciation qui permettent la spécialisation d'une de ces voies (*cf.* Figure 2.1). Ce modèle est soutenu par plusieurs études. Par exemple, il a été montré chez l'espèce $Escherichia\ coli$ que deux protéines homologues sont deux fois plus susceptibles d'être impliquées dans deux voies différentes que lorsqu'elles sont impliquées dans la même voie [Apic et al., 2001].

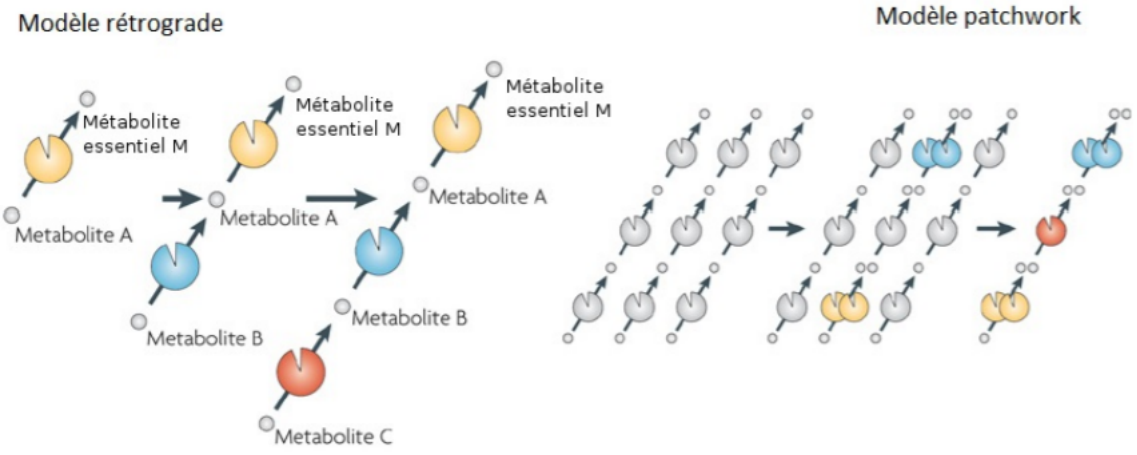


FIG. 2.1 : Deux modèles d'évolution du métabolisme : le modèle rétrograde à gauche et le modèle patchwork à droite (Figure de [Pereira, 2018]).

2.2.2 Le génome et ses relations avec le métabolisme

Après avoir présenter le métabolisme dans la section ci-dessus, nous présentons dans cette section le génome et ses relations avec le métabolisme puis nous présentons la production des protéines par les gènes. Le génome est l'ensemble de l'information génétique d'un organisme codée dans son ADN (ou ARN pour les virus). Il consiste dans le regroupement des différents chromosomes de l'organisme. La science qui l'étudie est la génomique. L'expression des gènes produit des protéines. Certaines d'entre elles jouent le rôle d'enzymes et catalysent des réactions biochimiques. Le comportement métabolique d'un organisme est ainsi contrôlé par l'expression des gènes présents dans son génome. Cependant, il est faux de considérer le génome comme indépendant du métabolisme, l'expression des gènes pouvant être déclenchée par des éléments constituants du métabolisme. Il est donc important d'étudier le gé-

nome et le métabolisme en association constante, l'un et l'autre étant indissociables dans un système vivant. Ces relations peuvent être représentées dans un diagramme UML. Les objets correspondent à des metabolites, des réactions biochimiques, des enzymes et des gènes (*cf.* Figure 2.2).

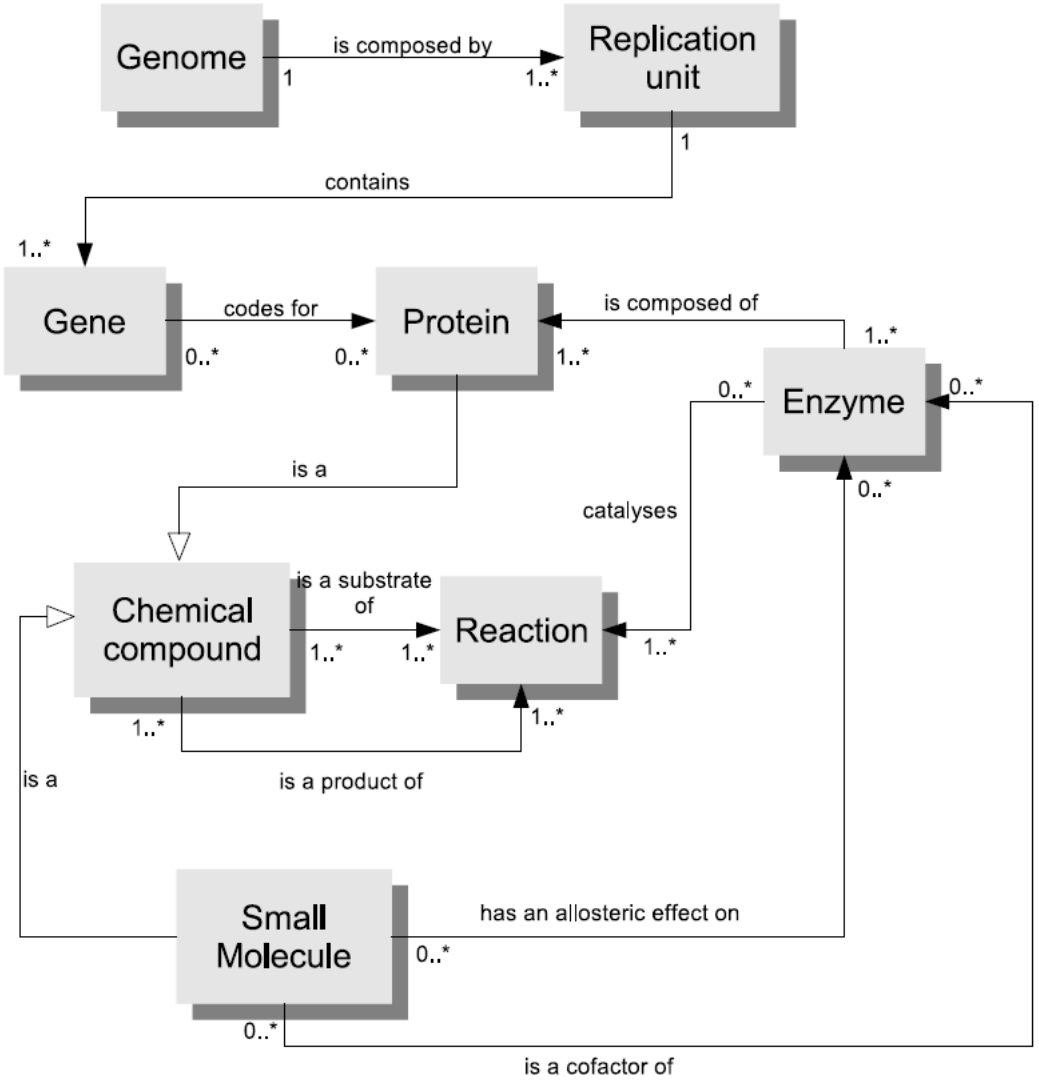


FIG. 2.2 : Schéma UML simplifié des différents objets impliqués dans le métabolisme [Lacroix et al., 2008].

2.2.2.1 Les chromosomes

Le génome est constitué d'un ou plusieurs chromosomes. Le chromosome (du grec khroma, couleur et soma, corps, élément) est l'élément porteur de l'information génétique. C'est un élément microscopique, contenu dans chaque cellule, qui est constitué de molécules d'acide désoxyribonucléique (ADN) formant une chaîne double brin linéaire ou circulaire. Un chromosome est souvent représenté dans la littérature sous sa forme compacte, semblable à un X. Cette forme se trouve chez les organismes appartenant à la famille des eucaryotes (cellule possédant un noyau). Pour les procaryotes (cellules ne possèdant pas de noyau), les chromosomes ont la forme de filaments (chaîne linéaire) ou d'anneaux (chaîne circulaire) (*cf.* Figure 2.3). Ils sont constitués de macromolécules appelées acide désoxyribonucléique,

ou ADN.

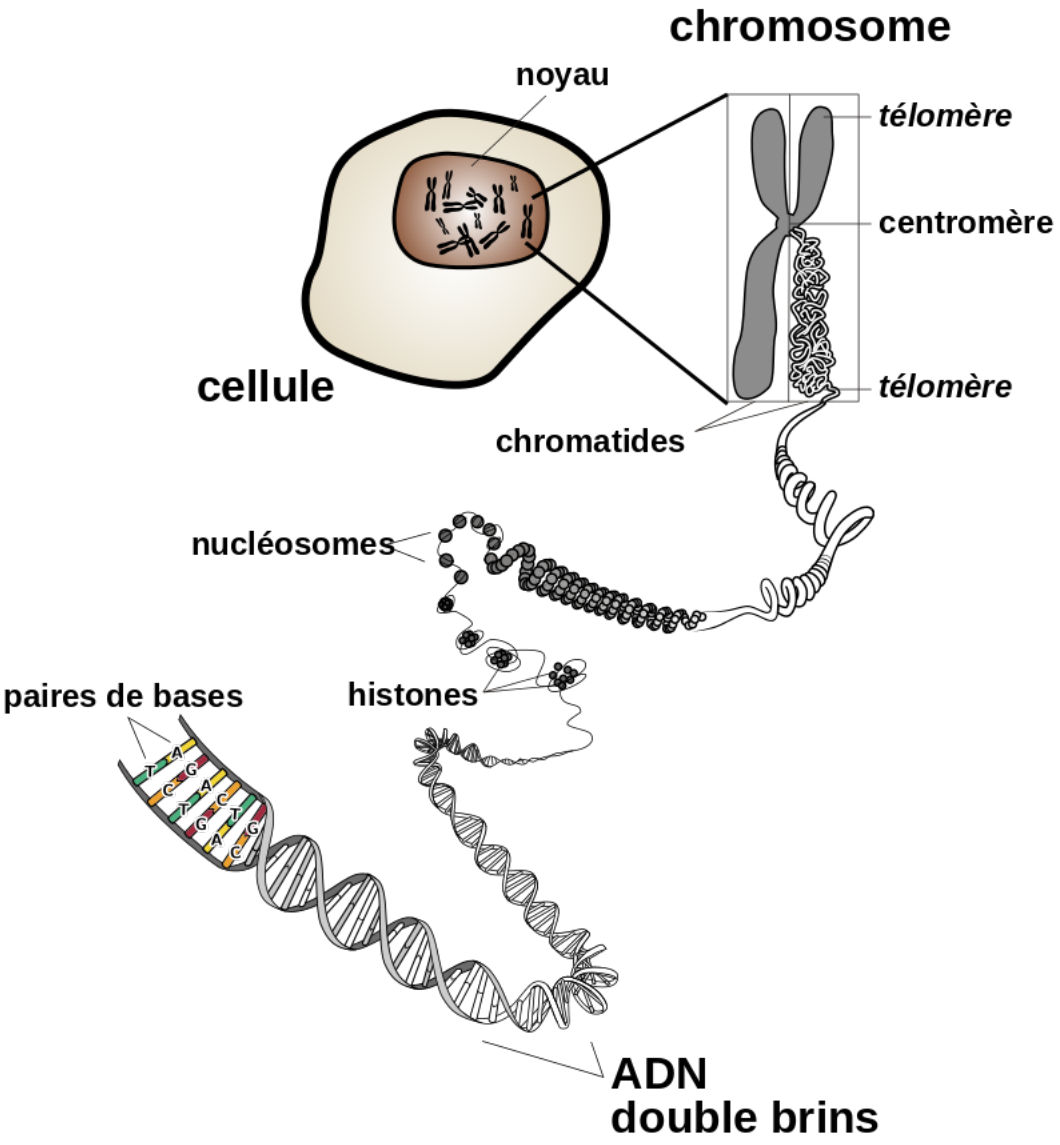


FIG. 2.3 : Illustration de la structure d'un chromosome (KES47, VIA WIKIMEDIA COMMONS CC 3.0).

2.2.2.2 L'ADN

L'ADN est constituée de deux brins complémentaires enroulés en double hélice. Chaque brin est formé d'une séquence de nucléotides. Un nucléotide est constitué d'un groupe phosphate, d'un désoxyribose et d'une base azotée. Il existe quatre bases azotées : l'adénine (A), la cytosine (C), la guanine (G) et la thymine (T). Ces bases sont complémentaires. L'adénine s'associe à la thymine ($A = T$) à l'aide d'une double liaison hydrogène, et la cytosine à la guanine ($C = G$) à l'aide d'un triple liaison hydrogène (*cf.* Figure 2.4).

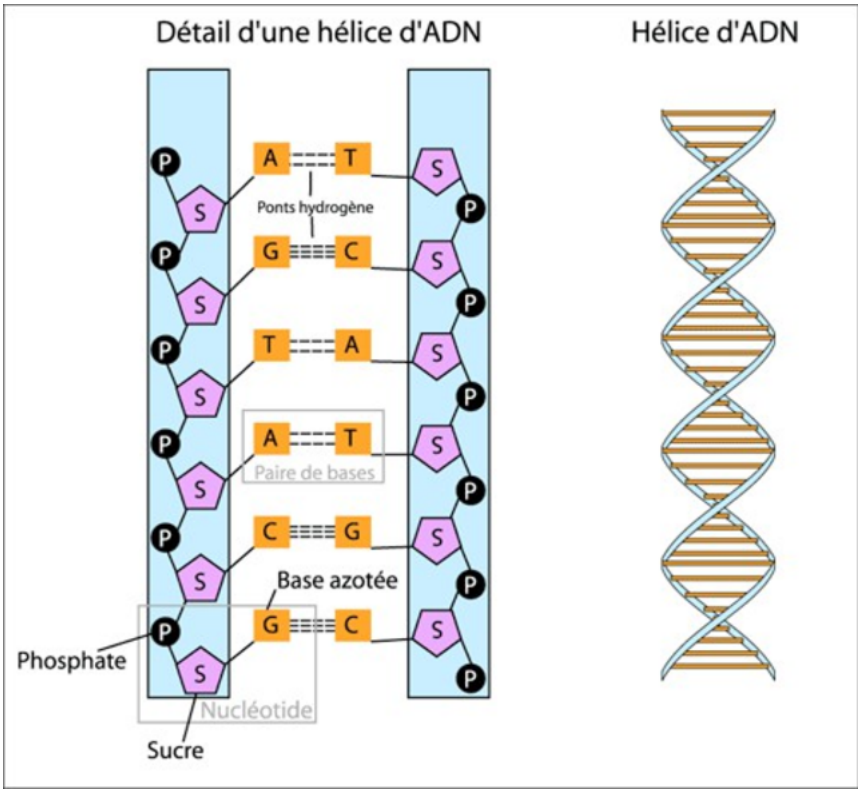


FIG. 2.4 : llustration de la structure d'ADN. (CC BY-SA 4.0, via Wikimedia Commons).

2.2.2.3 ARN

L'ARN (acide ribonucléique) est similaire à l'ADN dans sa composition. Il est composé de ribonucléotides dont la molécule de sucre à cinq carbones est le ribose (au lieu du désoxyribose). La base azotée thymine de l'ADN est remplacée par l'uracile (U) dans l'ARN. Il existe plusieurs types d'ARN, les trois plus courants étant l'ARN messager (ARNm), l'ARN de transfert (ARNt) et l'ARN ribosomal (ARNr).

Les molécules d'ADN sont très grandes par rapport à l'ARN. L'ARN, en revanche, est une molécule courte par rapport à l'ADN. Existant souvent sous forme de un seul brin, l'ARN est donc libre de se replier sur lui-même et d'adopter des formes tridimensionnelles qui servent divers rôles fonctionnels.

2.2.2.4 Les protéines

Nous venons de voir que l'expression des gènes permettait la synthèse de protéines. Une protéine est constituée d'une séquence d'acides aminés formant un (poly)peptide. Par attraction chimique cette forme linéaire se replie sur elle-même, ce qui donne une forme tridimensionnelle à la protéine et lui permet d'assurer sa fonction biochimique. Les enzymes constituent une classe particulière de protéines. Chacune d'entre elles est capable de catalyser des réactions chimiques, c'est-à-dire d'accélérer la transformation de composés chimiques en d'autres. La fonction de l'enzyme dépend des réactions qu'elle catalyse. L'ensemble des protéines générées par un organisme constitue son protéome, et la science qui étudie les protéines s'appelle la protéomique. Nous avons vu plus haut que ces enzymes jouaient un rôle essentiel du métabolisme.

2.2.2.5 Les gènes

Un gène se trouve sur un chromosome. C'est une séquence d'ADN qui spécifie la synthèse des protéines. Les gènes ont généralement une longueur de plusieurs centaines ou milliers de paires de bases car ils codent des protéines composées de centaines ou de milliers d'acides aminés.

Quelques gènes produisent d'autres molécules qui aident la cellule à assembler les protéines. Le passage du gène à la protéine est complexe et étroitement contrôlé dans chaque cellule. Il se compose de deux grandes étapes : la transcription et la traduction. Ensemble, la transcription et la traduction sont connues sous le nom d'expression génétique.

Le processus de transcription a lieu dans le noyau de la cellule. Au début, la double hélice de l'ADN se sépare, permettant ainsi la formation d'un brin d'ARNm complémentaire à l'un des deux brins d'ADN. Des paires complémentaires se forment entre l'ADN et l'ARNm : la cytosine avec la guanine et, à la différence des paires complémentaires de l'ADN, l'adénine se lie avec l'uracile. Une fois la molécule d'ARNm complétée, elle se détache et traverse à l'extérieur du noyau, dans le cytoplasme. La prochaine étape est la traduction de cette molécule d'ARNm.

Une fois la transcription terminée, l'ARNm libérée dans le cytoplasme va rejoindre un ribosome, organite responsable de la traduction. Un deuxième type d'ARN est requis pour faire la traduction, il s'agit de l'ARN de transfert, aussi nommée ARNt. Cette molécule a deux extrémités ayant chacune leur fonction. La première est chargée du transport de l'acide aminé désignée par le codon de l'ARNm. La deuxième comporte un triplet complémentaire au codon de l'ARNm.

Pendant la traduction, le ribosome défile le long du brin d'ARNm en exposant un à un les codons de celui-ci. Ainsi, les molécules d'ARNt apportent les acides aminés dans l'ordre indiqué par la séquence d'ARNm. Le ribosome se charge ensuite “d'attacher" les acides aminés ensemble pour ainsi former une chaîne polypeptidique, c'est-à-dire une protéine.

Le flux d'informations de l'ADN à l'ARN et aux protéines est l'un des principes fondamentaux de la biologie moléculaire connu sous la nomenclature “dogme central" (*cf.* Figure 2.5).

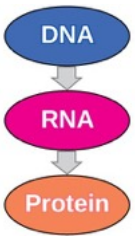


FIG. 2.5 : Le *dogme central* de la biologie moléculaire. L'ADN sert à fabriquer l'ARN, qui sert à fabriquer les protéines.

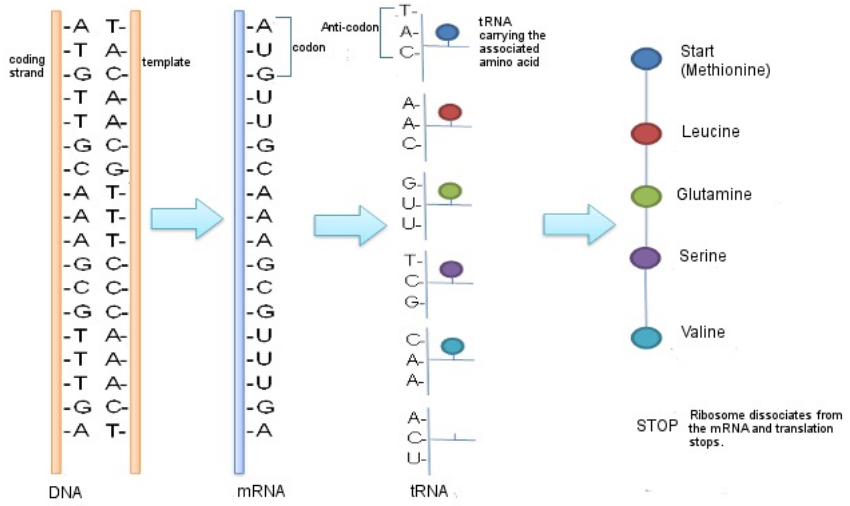


FIG. 2.6 : Détails sur le *dogme central.* (CC BY-SA 2.0, , via Wikimedia Commons).

Un groupe de gènes co-localisés qui sont co-régulés et co-transcrits est appelé opéron. Les gènes des opérons ont tendance à être associés à une fonction biologique donnée [Overbeek et al., 1999]. Il a été estimé qu'environ 60% des gènes chez E. coli sont organisés en opérons [Kevans et al., 2015]. En général, les opérons sont bien conservés entre les espèces, bien que certains gènes puissent être réarrangés, gagnés/perdus, ou dupliqués [Maguire et al., 2015].

Comme nous l'avons vu en Introduction, les scientifiques se sont longtemps focalisés sur l'étude de séquences pour comprendre le fonctionnement des composants biologiques que nous venons d'évoquer (séquences du nucléotides pour l'ADN, séquences d'acides aminées pour les protéines, etc.). Ils ont donc, de façon naturelle, chercher à comparer ces séquences pour comprendre et extraire les similitudes et les différences qui peuvent exister entre les espèces. Nous allons maintenant présenter différentes techniques permettant de comparer des séquences.

2.3 Alignements des séquences

Dans cette section, nous évoquons la notion de quantification et de qualification de la similarité entre séquences. Aujourd'hui l'alignement de séquences est le moyen de comparaison de séquences le plus utilisé. L'alignement des séquences permet de caractériser une sequence inconnue par sa similarité avec d'autres séquences connues. Par conséquent des nouvelles fonctions biologiques (ADN, ARN, protéine, ...) peuvent être reconnues [Pearson and Lipman, 1988, Altschul et al., 1990, Myers, 1991, Waterman, 1998, Loweth, 1997, Gusfield, 1997]. Ces informations fournissent des données supplémentaires sur la fonctionnalité, l'originalité ou l'évolution des espèces où ces séquences biologiques sont obtenues. La comparaison des séquences permet d'observer les régions conservées et d'évaluer les liens ancestraux entre les séquences étudiées. L'opération de comparaison peut être définie de manière très simplifiée comme l'identification des parties similaires des séquences étudiées. Si le concept est simple, sa mise en oeuvre l'est beaucoup moins, simple car un critère de similarité doit être formellement défini.

Les séquences peuvent être comparées localement ou globalement. La procédure classique lors de la comparaison de séquences consiste à examiner les similitudes entre les séquences globalement, c'est-à-dire à considérer les séquences dans leur ensemble. Ce système de comparaison fonctionne bien tant que les séquences comparées se ressemble effectivement. Dès que l'on veut comparer des séquences qui ne sont pas assez proches les unes des autres, une méthode basée sur la similarité globale ne permettra

pas d'obtenir des informations pertinentes. En effet, lorsqu'on compare globalement deux séquences relativement distantes, la méthode risque de forcer des parties des deux séquences peu similaires à correspondre. Un autre processus de comparaison, appelé alignement local consiste à rechercher des portions des deux séquences présentant une bonne similitude. Il s'agit ensuite d'identifier la similitude locale entre les deux séquences. Ces méthodes fournissent souvent plusieurs résultats, accompagnés d'un score qui reflète l'intérêt ou la qualité de la similarité locale.

Nous allons présenter ci-dessous les deux méthodes les plus connues pour la comparaison des séquences, FastA et BLAST.

2.3.1 FastA : Fast Alignement

Pearson $et\ al.$ [Pearson and Lipman, 1988] ont développé un algorithme basé sur l'identification rapide des zones similaires entre la séquence étudiée et les séquences de la base de données. Cette reconnaissance permet de considérer uniquement les séquences présentant une région de forte similitude avec la séquence recherchée. Un algorithme d'alignement optimal est appliqué sur la meilleure zone de ressemblance. Pour chaque séquence de la banque, l'algorithme se déroule en quatre étapes distinctes.

- La première étape consiste à repérer les régions les plus denses en entités partagées par les deux séquences. La longueur du mot correspondant est appelée le paramètre $ktup$.
- Dans une deuxième étape, une matrice de scores élémentaires est construite entre les dix régions ayant obtenu les meilleurs scores lors de la phase 1. Les scores obtenus correspondent à des régions initiales de premier ordre et l'on qualifie de score $init_1$ celui qui représente la région de plus fort score parmi les dix analysées.
- La troisième étape essaie de joindre les régions définies à l'étape précédente, bien entendu s'il en existe au moins deux et si chacune de celles-ci possède un score supérieur à un score seuil prédéfini. Ce seuil correspond en fait à un score moyen attendu pour des séquences non apparentées. On réunira ces régions initiales à chaque fois que la somme de leur scores diminuée d'une pénalité de jonction est supérieure ou égale au score $init_1$. Ce score s'il existe est appelé $init_n$ et correspond à une région initiale de deuxième ordre.
- La quatrième étape consiste à effectuer l'alignement optimal de la séquence recherchée avec la séquence de la banque en considérant uniquement les parties des séquences délimitées par la meilleure région initiale de score $init_n$. On obtient alors un score optimal dénommé opt. Cet alignement est effectué uniquement pour un nombre limité de séquences fixé par l'utilisateur. Ce sont les séquences qui correspondent aux plus hauts scores initiaux $init_n$.

2.3.2 BLAST : Basic Local Alignment Search Tool

Altschul $et\ al.$ [Altschul et al., 1990] ont inventé un algorithme utilisé pour comparer les séquences d'acides aminés de différentes protéines ou les séquences de nucléotides d'un acide nucléique. Grâce à une recherche BLAST, il est possible de trouver toutes les séquences de la base qui sont similaires à une sequence donnée S.

L'algorithme peut être décomposé comme suit :

- L'algorithme va dans un premier temps chercher toutes les sous-séquences de S de taille w. Par défaut, w prends la valeur 3 pour la comparaison de séquences des protéines et la valeur 11 pour l'ADN. L'algorithme crée alors une liste des sous-séquences de S dont le score d'alignement est supérieur à une valeur fixée.
- La deuxième étape consiste à exploiter la liste trouvée dans la première étape, ainsi que celles de la base de données qui lui sont liées. Pour chaque paire ainsi formée, l'algorithme cherche à étendre au maximum la longueur des sous-séquences. Cette extension est limitée par un seuil fixé.
- Dans cette dernière étape, BLAST ordonne les résultats obtenus en fonction de leur score de similarité.

Malheureusement, un système biologique ne peux être compris en se basant uniquement sur la fonction de chaque composant biologique. D'après [Aderem, 2005], la compréhension des systèmes biologiques nécessite la prise en considération de trois concepts clés, à savoir : *modularité, robustesse* et *émergence*. Les concepts de *modularité* et de *robustesse* décrivent la capacité de ces systèmes à évoluer et à résister aux perturbations, respectivement. En outre, ces systèmes possèdent des propriétés qui ne peuvent être expliquées par aucune de leurs parties individuelles. Ces propriétés dites *emergentes* sont le résultat de leurs relations et interactions. Cela explique pourquoi la recherche s'est déplacée de l'étude des composants individuels (par exemple, en comparant les séquences de gènes, de protéines ou de métabolites) vers l'étude des relations entre les composants en comparant les réseaux biologiques [Bruggeman and Westerhoff, 2007, Towfic et al., 2009, Shyu and Tsai, 2009, Mousavi and Tabataba, 2012, Blazewicz et al., 2018, Evans et al., 2011, Baewicz et al., 2005]. La recherche sur ces réseaux a été très active ces dernières années, la comparaison des réseaux, en particulier, devenant de plus en plus pertinente. Dans les sections suivantes, nous présentons des méthodes de la comparaison des réseaux homogènes et de réseaux hétérogènes.

2.4 Topologie des réseaux

La topologie des réseaux est un domaine extrêmement vaste. Les réseaux possèdent certaines propriétés qui sont très utiles pour découvrir les informations qu'ils contiennent. Le but de tout type d'analyse de réseau est d'extraire des informations significatives. L'extraction de ces informations est impossible si les composants individuels étaient examinés séparément.

La topologie est la manière dont les noeuds et les arêtes sont disposés dans un réseau. Les propriétés topologiques peuvent s'appliquer au réseau dans son ensemble ou à des noeuds et des arêtes individuels. Une description plus complète sur la topologie des réseaux peut être trouvée [Boccaletti et al., 2006]. Les propriétés et les concepts topologiques les plus utilisés sont détaillés dans la section suivante.

2.4.1 Caractéristiques

Il existe de nombreux outils et mesures disponibles pour étudier la structure des réseaux complexes. Dans ce qui suit, nous allons aborder les mesures les plus fondamentales pour étudier la topologie des réseaux.

2.4.1.1 Distribution des degrés

La distribution des degrés est la probabilité $p(k)$ qu'un noeud choisi au hasard ait un degré k donné. L'une des principales révélations qui a donné de l'intérêt pour la théorie des réseaux complexes est que la distribution $p(k)$ de nombreux réseaux suit approximativement une loi de puissance $p(k) \sim k^{-\gamma}$, où γ est l'exposant du degré. La grande majorité des sommets ne possèdent qu'un petit nombre de voisins, un petit nombre de sommets (les hubs) sont fortement connectés [Barabasi and Albert, 1999] (voir Figure2.7). Parmi les exemples importants, citons le réseau métabolique de 43 organismes [Jeong and Albert, 2000], le réseau d'interaction des protéines de S. Cerevisiae [Jeong et al., 2001]. Il existe d'autres réseaux qui subissent un processus de formation purement aléatoire et, par conséquent, tous les noeuds ont un degré similaire. La distribution aléatoire $P(k)$ suit une loi de Poisson. Dans ce cas $P(k)$ est maximum pour une valeur moyenne k_{moyen}, et décroît quand $k < k_{moyen}$ ou $k > k_{moyen}$ (voire Figure2.7).

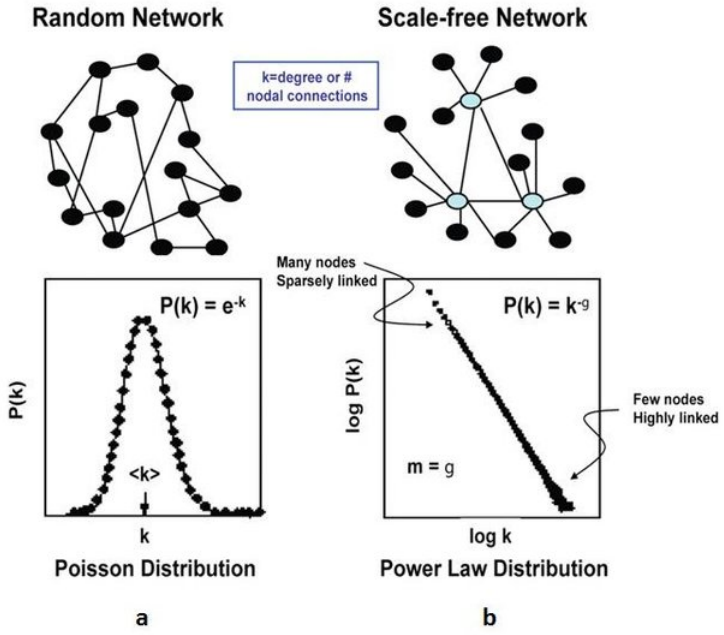


FIG. 2.7 : Loi de Poisson et loi de puissance. (Source [Chan and Loscalzo, 2012]).

2.4.1.2 Coefficient de clustering d'un graphe

Le coefficient de clustering est une mesure qui donne un aperçu de la structure locale d'un réseau. Plus précisément ce coefficient mesure à quel point le voisinage d'un sommet est connecté [Rubinov and Sporns, 2010]. Pour un noeud i avec un degré k_i, le coefficient de clustering est défini par $C_i = \frac{2n_i}{k_i(k_i - 1)}$, représentant le rapport entre le nombre de connexions entre les voisins du sommet i et le nombre connexions possibles.

2.4.1.3 Les sous graphes et motifs de réseaux réels complexes

Les motifs représentant des patterns d'interaction élémentaires entre petits groupes de sous-graphes [Albert and Barabási, 2002, Strogatz, 2001]. Les preuves théoriques et expérimentales indiquent qu'au moins certains de ces patterns d'interaction élémentaires sont porteurs d'informations significatives sur la fonction et l'organisation globale du réseau [Shen-Orr et al., 2002, Milo et al., 2002, Milo et al., 2004].

2.4.1.4 La centralité des noeuds dans un réseau

La centralité est l'un des nombreux concepts qui ont été créés dans l'analyse des réseaux sociaux, puis importés dans l'étude de tout type de système en réseau [Arita, 2004]. Nous décrivons ici certaines mesures de centralité pertinentes actuellement utilisées pour l'étude des réseaux complexes.

Les mesures de centralité tentent de saisir la notion d'"importance" des noeuds dans les réseaux en quantifiant la capacité d'un noeud à communiquer directement avec d'autres noeuds, ou sa proximité avec de nombreux autres noeuds, ou le nombre de paires de noeuds qui ont besoin d'un noeud spécifique comme intermédiaire dans leurs communications. La centralité intermédiaire permet de déterminer dans quelle mesure un noeud donné (noté k) se trouve entre d'autres noeuds. Cette métrique est mesurée par le nombre de chemins les plus courts (entre deux noeuds quelconques du graphe) qui

passent par le noeud cible k (noté $\sigma_{i,j}(k)$). Ce score est modéré par le nombre total de plus courts chemins existant entre n'importe quel paire de noeuds du graphe (noté $\sigma_{i,j}$). Le noeud cible aura un score de centralité élevée s'il apparaît dans de nombreux chemins les plus courts.

$B(k) = \sum_{i \neq j \neq k} \frac{\sigma_{i,j}(k)}{\sigma_{i,j}}$

Dans la Figure 2.8, le sommet k a une centralité élevée car tous les chemins les plus courts entre les sommets des sous-réseaux bleu et rouge passent par k.

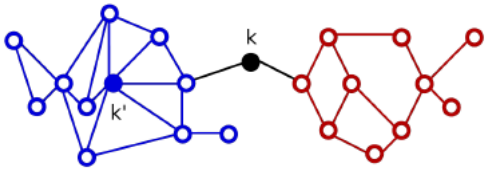


FIG. 2.8 : Centralité intermédiaire. Le degré d'un sommet ne reflète pas nécessairement son importance dans le réseau. Bien que le sommet k' ait un degré élevé, sa suppression n'affecterait pas la communication au sein du réseau. Cependant, comme le sommet k a une centralité élevée, sa suppression aurait un impact sur la communication [Steuer and Lopez, 2008].

2.4.2 Modèles des réseaux

La série d'articles publiés par Erdös et Rényi sur les graphes aléatoires, entre 1950 et 1960, a suscité un intérêt initial pour la science des réseaux. Cependant, depuis l'introduction des réseaux à petit monde par [Watts and Strogatz, 1998], l'intérêt pour la science des réseaux s'est accru, comme le montre la littérature. Les réseaux suivants sont désormais largement considérés comme des références ;

2.4.2.1 Modèle aléatoire

L'étude des graphes aléatoires est l'un des domaines les plus importants en théorie des graphes. Les graphes aléatoires ont trouvé de multiples applications en physique et ils sont utilisés aujourd'hui comme modèle standard dans la simulation de nombreux processus physiques sur les graphes et les réseaux. Il existe plusieurs façons de définir un graphe aléatoire, c'est-à-dire un graphe dans lequel, étant donné un ensemble de noeuds, les arêtes qui les relient sont aléatoires. Le modèle le plus simple de graphe aléatoire a été introduit par Erdös et Rényi [Renyi, 1959]. La construction d'un graphe aléatoire dans ce modèle commence par considérer n noeuds isolés. Chaque paire de noeuds est reliée avec une probabilité $p > 0$. Par conséquent, le graphe est déterminé uniquement par le nombre de noeuds et d'arêtes $G(n,m)$ ou $G(n,p)$. La Figure (2.9) illustre quelques exemples de graphes d'Erdös-Rényi. Nous illustrons quelques exemples de graphes aléatoires de Erdös-Rényi avec le même nombre de noeuds et différentes probabilités de liaison.

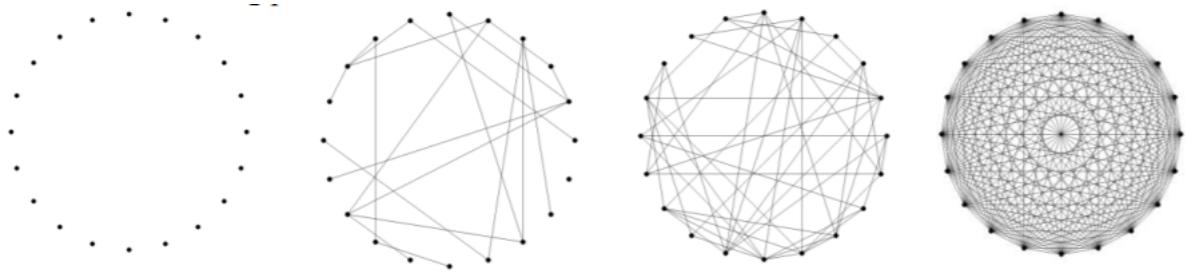


FIG. 2.9 : llustration des changements d'un réseau aléatoire Erdös-Rényi avec 20 noeuds et des des probabilités qui augmentent de zéro (à gauche) à un (à droite). (CC BY 4.0, via Wikimedia Commons).

2.4.2.2 Modèle du petit monde

L'un des concepts les plus répandus dans la théorie des réseaux est celui du "petit monde". Un graphe k-régulier (tous les sommets ont un même degré k) sur lequel s'appuie le modèle de Watts $et\ al.$ [Watts and Strogatz, 1998] est un anneau de N sommets, où chaque sommet est connecté à ses k plus proches voisins. Ensuite, avec une probabilité p, l'une des extrémités de chaque arête est redirigée vers un sommet quelconque du graphe (*cf.* Figure 2.10). Le modèle de Watt-Strogatz commence par utiliser la construction suivante. Placez tous les noeuds dans un cercle et connectez chaque noeud à ses $k/2$ premiers voisins les plus proches dans le sens des aiguilles d'une montre, ainsi qu'à ses $k/2$ voisins les plus proches dans le sens inverse des aiguilles d'une montre (*cf.* Figure 2.10). Le coefficient de regroupement moyen pour ces réseaux est donné par [Barrat and Weigt, 2001]
$C = \frac{3(k\quad -2)}{(k\quad -1)}$, ce qui signifie que $C = 0.75$ pour chaque valeurs de k.

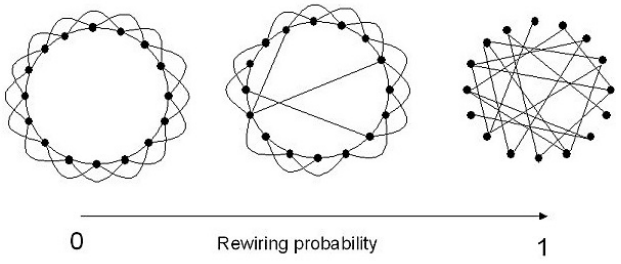


FIG. 2.10 : Représentation schématique de l'évolution du processus de recâblage dans le modèle de WattsStrogatz. (CC BY 4.0, via figshare).

2.4.2.3 Modèle de Scale-free

Barabási et Albert ont introduit deux mécanismes clés absents du modèle de réseau aléatoire classique. Ces deux mécanismes sont responsables de l'émergence d'une distribution des degrés en loi de puissance [Barabasi and Albert, 1999]. Tout d'abord, les réseaux se développent par l'ajout de nouveaux noeuds se reliant aux noeuds déjà présents dans le système. Deuxièmement, avec la propriété appelée attachement préférentiel, il y a une plus grande probabilité de se lier à un noeud ayant un grand nombre de connexions. Ces deux principes sont mis en oeuvre comme suit : à partir d'un petit graphe de base composé de m_0 noeuds, un nouveau noeud avec m liens est ajouté à chaque itération et connecté aux noeuds déjà existants (*cf.* Figure 2.11). Chacun des m nouveaux liens est ensuite attaché de manière préférentielle à un noeud i (avec ki voisins) qui est choisi en fonction de l'état de la probabilité $\Pi_i = \quad k_i / \sum_{k_j}$.

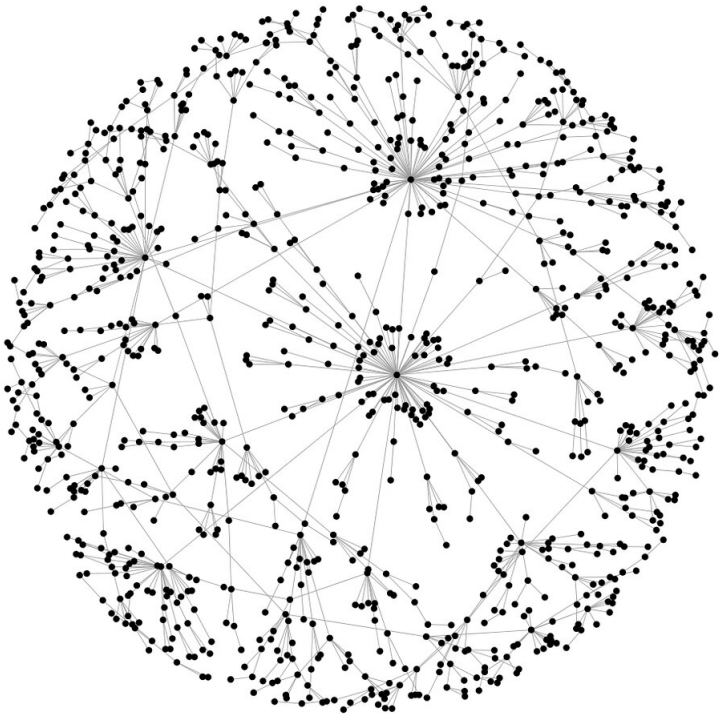


FIG. 2.11 : Exemple de réseau scale-free. (Source [Barabasi and Albert, 1999]).

2.4.2.4 Modèle hiérarchique

Ravasz *et al.* [Ravasz et al., 2002] ont affiné le modèle scale-free proposé par Barabási et Albert [Barabasi and Albert, 1999] en un modèle hiérarchique présentant la même topologie scale-free et une modularité intégrée (*cf.* Figure 2.12). Ce modèle est caractérisé par le fait que le coefficient de clustering d'un noeud de degré k décroît sous la forme $C(k) \sim k^{-1}$. Les auteurs ont validé leur modèle en mesurant le coefficient de clustering dans les réseaux métaboliques de 43 organismes. L'étude suggère que les réseaux métaboliques contiennent plusieurs grands modules qui, à leur tour, sont constitués de sous-modules plus petits.

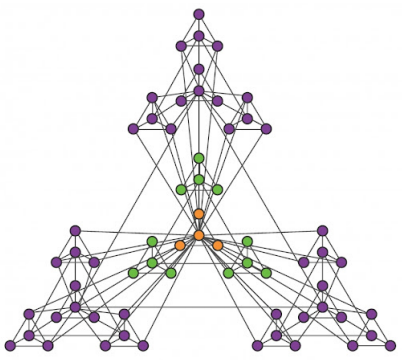


FIG. 2.12 : Exemple de modèle hiérarchique.(Source [Barabasi and Oltvai, 2004]).

2.5 Les réseaux biologiques et leurs modélisation

Un réseau biologique est une représentation abstraite d'un système biologique. La structure de ce réseau est modélisée par des graphes. Les graphes sont communément utilisés comme support formel et leur sémantique peut être diverse. Dans le cas d'un réseau d'interactions biologiques, les composants biologiques (*e.g.* les protéines) sont représentés par des sommets, et une arête entre deux sommets représente une possible interaction. Selon la nature de ces réseaux, les graphes correspondants sont orientés ou non-orientés. Nous présentons ci-après le réseau d'interaction protéine-protéine et le réseau métabolique.

2.5.1 Réseau d'interaction protéine-protéine

L'ensemble des interactions protéine-protéine ayant lieu dans une cellule est souvent modélisé par un réseau dans lequel les sommets correspondent à des protéines, et où une arête entre deux sommets signifie l'existence d'une interaction entre les deux protéines correspondantes [Koyutürk et al., 2004, Klau, 2009, Towfic et al., 2009, Sharan et al., 2007, Liao et al., 2009, Istrail et al., 2012, Ciriello and Guerra, 2008] . Il existe des méthodes expérimentales pour l'identification de ces interactions. Deux méthodes sont principalement utilisées, la première est appelée double hybride [Fields and Song, 1989, Ito et al., 2001] qui permet d'identifier les interactions entre deux protéines, la deuxième est appelée la spectrométrie de masse [Fields and Song, 1989] qui permet d'identifier les interactions entre différentes protéines. Il est important de noter que ces méthodes ne sont pas exactes, parfois elles donnent des interactions qui n'existent pas vraiment, que l'on appelle faux positifs ; des interactions réelles peuvent également manquer (faux négatifs). La Figure 2.13 illustre un exemple de réseau d'interaction protéine-protéine chez la levure du boulanger [Barabã et al., 2002].

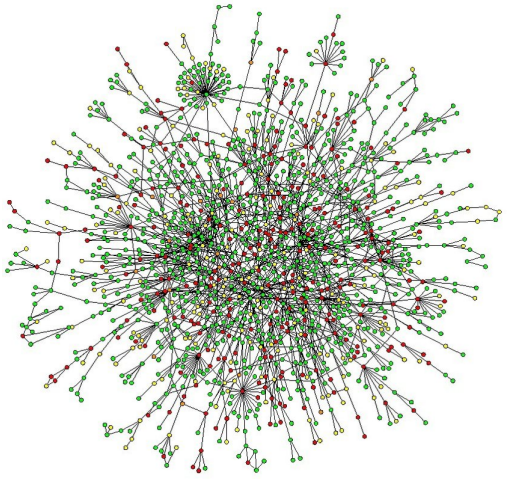


FIG. 2.13 : Visualisation d'un réseau d'interaction protéine-protéine chez la levure du boulanger [Barabã et al., 2002].

Étant donné que plusieurs études [Cohen et al., 2001, Barratt et al., 1990] ont démontré que les réseaux d'interactions protéines/protéines sont de type scale-free, nous évaluons les méthodes proposées sur des données aléatoires.

2.5.2 Réseau métabolique

Le réseau métabolique peut être découpé en un ensemble de sous-réseaux appelés voies métaboliques. Chaque voie métabolique est une succession de réactions biochimiques permettant de transformer une molécule en une autre. Le découpage du réseau métabolique en différentes voies métaboliques est important pour en simplifier la compréhension. Chaque sous-réseau constitue alors un processus biologique jouant un rôle fonctionnel. Ce découpage est en grande partie effectué à la main. C'est un travail fastidieux qui a entraîné la proposition de méthodes de décomposition et de découpage automatique de celui-ci. La Figure 2.14 illustre une voie métabolique chez la bactérie *Escherichia coli*.

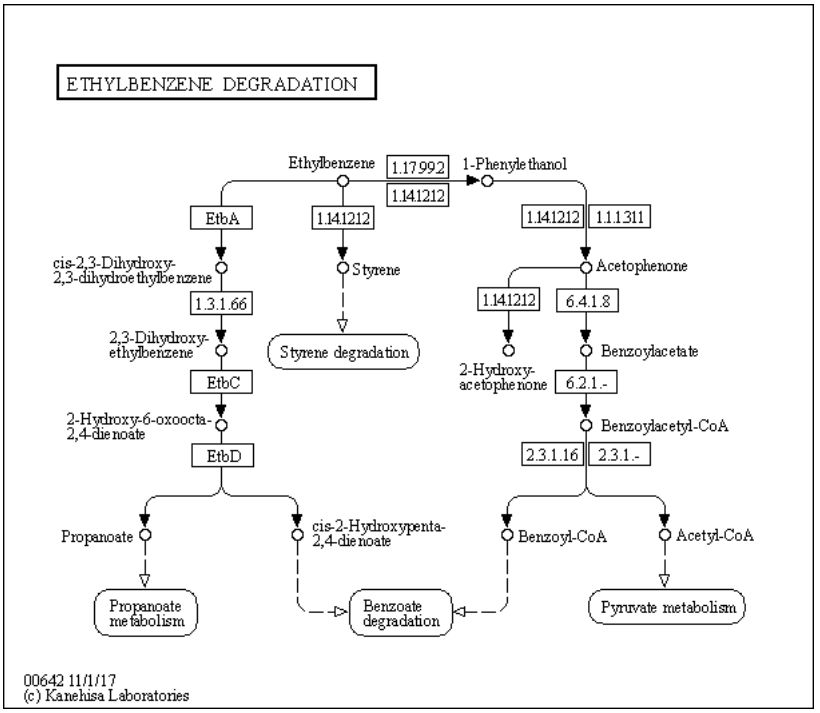


FIG. 2.14 : Exemple d'une voie métabolique : il s'agit d'un graphe biochimique, les sommets rectangulaires sont des enzymes, les sommets ronds sont des métabolites, les sommets rectangulaires avec des coins arrondis renvoient vers d'autres voies métaboliques. Image extraite de KEGG

Il existe essentiellement cinq types de représentations des réseaux métaboliques : graphe des composants (substrats et produits), graphe des réactions, graphe des enzymes et graphe biparti (composants vs réactions). Ces graphes peuvent être orientés ou non-orientés, l'orientation permettant de différencier. Le réseau métabolique peut être modélisé par différents graphes métaboliques [Lacroix et al., 2008] :

1. **Un graphe biparti**. C'est un graphe dans lequel les sommets peuvent être séparés en 2 groupes, tel que toutes les arêtes du graphe relient des sommets appartenant à des groupes distincts. Le graphe biparti permet de représenter à la fois les métabolites et les réactions sous forme de sommets. Un noeud métabolite est prédécesseur d'un sommet réaction s'il est substrat de cette réaction, et il est successeur d'un sommet réaction s'il est produit par cette dernière (*cf.* Figure 2.15).

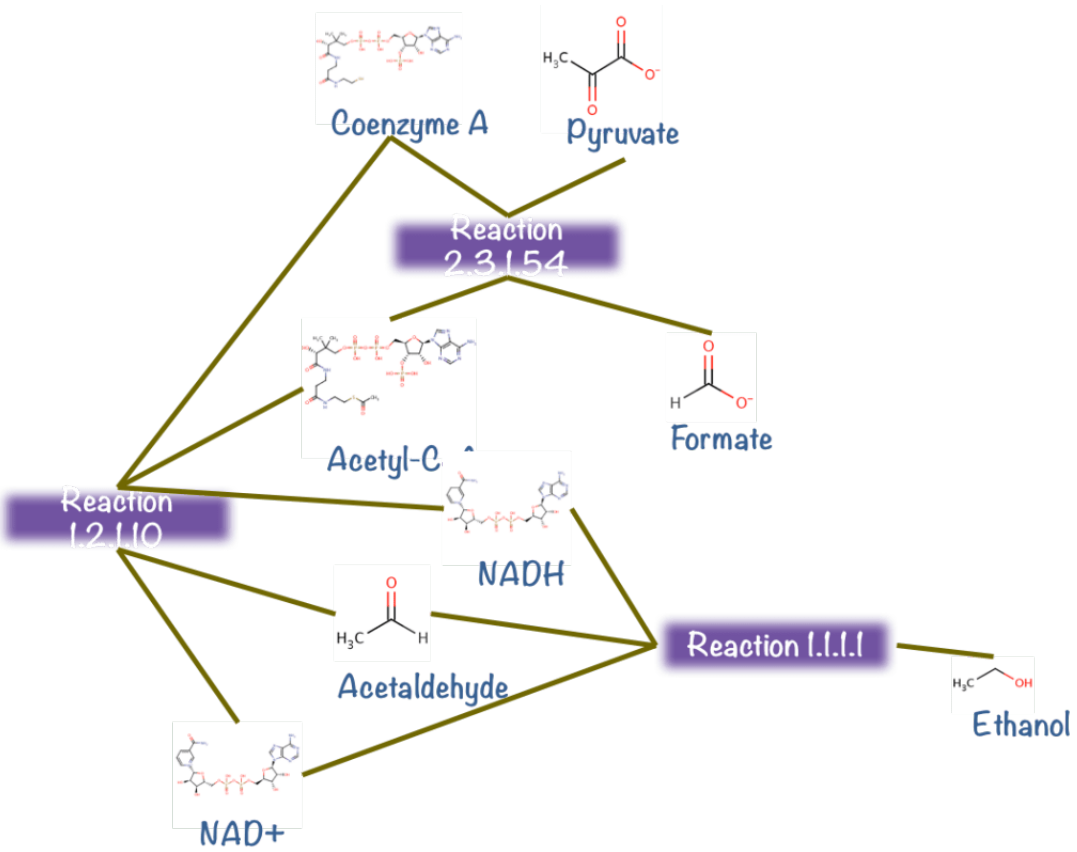


FIG. 2.15 : llustration d'un graphe biparti (image à partir de bioinfo-fr).

2. **Un hypergraphe**. C'est un graphe contenant des hyper-arcs. Un hyper-arc est une arc reliant plus de 2 sommets. Dans les réseaux métaboliques sous forme d'hypergraphes, les sommets représentent les métabolites, et les hyper-arcs représentent les réactions. Leurs origines correspondent aux substrats de la réaction, et leurs terminaisons correspondent aux produits (*cf.* Figure 2.16). Généralement, toutes les méthodes développées pour l'identification des modules biologiques ont été définies pour des graphes simples. De nouvelles méthodes qui s'appliqueraient à des graphes bipartis ou à des hypergraphes doivent probablement être développées [Lacroix et al., 2008].

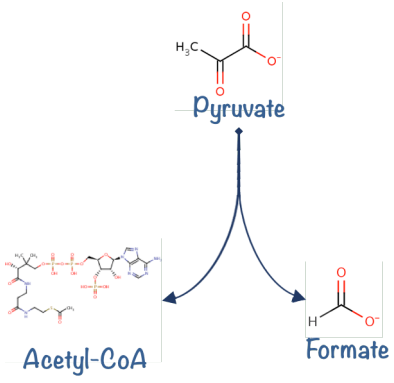


FIG. 2.16 : llustration d'un hypergraphe (image à partir de bioinfo-fr).

3. **Un graphe de métabolites**. C'est une forme réduite du réseau métabolique. Il se concentre sur les métabolites. C'est un graphe qui peut être orienté ou non-orienté, dont les sommets représentent les métabolites et une arête relie 2 métabolites s'il existe une réaction qui consomme l'un et produit l'autre (*cf.* Figure 2.17).

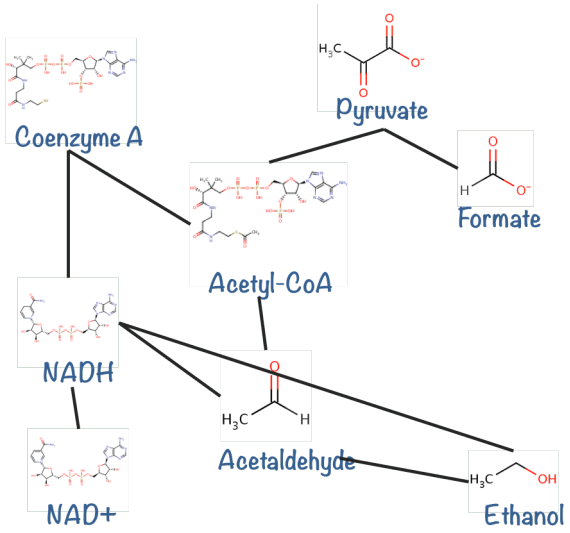


FIG. 2.17 : Ilustration d'un réseau de métabolites. (image à partir de bioinfo-fr).

4. **Un graphe de réactions**. Ce graphe est également une forme réduite du réseau métabolique. C'est un graphe orienté dans lequel les sommets représentent des réactions, et un arc relie deux réactions si l'une produit un métabolite qui est consommé par l'autre (*cf.* Figure 2.18).

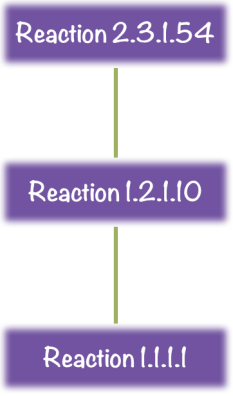


FIG. 2.18 : Ilustration d'un réseau de réactions. (image à partir de bioinfo-fr).

5. **Un graphe d'enzymes**. Dans ce graphe, les sommets correspondent aux enzymes. Elles sont reliées par une arête si elles catalysent des réactions qui ont un composé chimique en commun (*cf.* Figure 2.19).

FIG. 2.19 : Ilustration d'un réseau d'enzymes. (image à partir de bioinfo-fr).

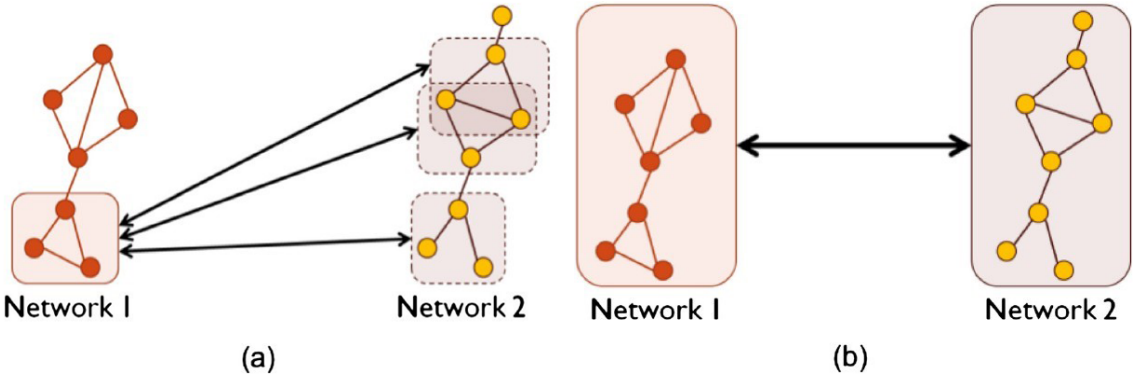
Dans cette thèse, nous travaillerons avec une modélisation de réseaux métaboliques par des graphes de réactions (modèle 4).

2.6 Alignements des réseaux

La recherche sur la comparaison de réseaux concerne principalement les représentations homogènes de données, telles que l'alignement de graphes orientés [Bunke, 2000], l'alignement de graphes non-orientés [Conte et al., 2004], et l'exploration de réseaux sociaux [Matsuo et al., 2006]. Ces recherches ont un large éventail d'applications; par exemple, la comparaison de réseaux d'interactions protéine-protéine pourrait améliorer notre compréhension des processus biologiques sous-jacents. La comparaison de réseaux homogènes peut également être utilisée pour étudier l'évolution des réseaux dans le temps et ainsi identifier les changements et les chocs soudains.

Aligner des réseaux consiste à les comparer à fin d'identifier des structures communes entre ceux-ci. L'alignement de réseaux a été utilisé dans de nombreux domaines et applications [Emmert-Streib et al., 2016]. En bio-informatique, l'alignement de réseau a été utilisée pour prédire la fonction des protéines, en alignant les réseaux d'interaction des protéines. Ce principe va permettre de remarquer des points communs et de mettre en évidence des fonctions biologiques entre deux réseaux ou plus. Les réseaux alignés peuvent être d'espèces différentes, ou bien d'une même espèce mais acquis à des instants différents ou sous différentes conditions [Elmsallati et al., 2018, Loch and Faisal, 2015, Guzzi and Milenković, 2018, Meng et al., 2016] ; L'alignement de réseaux a été utilisé aussi pour construire des arbres phylogénétiques d'espèces en se basant sur les similitudes de leurs réseaux d'interactions protéine-protéine ou de leurs réseaux métaboliques [Kuchaiev et al., 2010, Kuchaiev and Prulj, 2011].

De manière générale, le problème d'alignement des réseaux est $\mathcal{NP}$-complet La plupart des méthodes proposées pour résoudre ce problème sont des heuristiques. L'alignement de réseaux peut être classé en deux catégories : alignement local et alignement global [Ma and Liao, 2020]. La différence entre ces deux types est similaire à celle faite pour l'alignement de séquences. L'alignement global de réseaux tente de faire correspondre différents réseaux dans leur ensemble, et le résultat de sortie est une correspondance unique entre les sommets des réseaux. Il vise à identifier la meilleure correspondance cohérente entre tous les sommets des réseaux, ce qui peut révéler des fonctions conservées par l'évolution au niveau des systèmes. En revanche, l'alignement local des réseaux fait correspondre les sous-réseaux partiels entre les réseaux, ce qui permet difficilement de montrer la trace évolutive de systèmes entiers comme le montre la Figure 2.20(a). Il existe des méthodes pour l'alignement par paire et d'autres pour l'alignement multiple des réseaux.

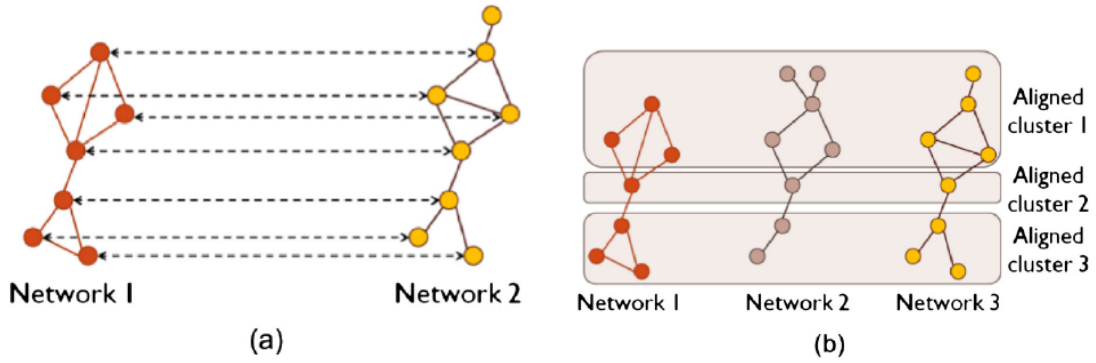


(a) Alignement local.

(b) Alignement global.

FIG. 2.20 : Alignement local et alignement global. Reproduit de [Loch and Faisal, 2015].

Pour les deux approches, locale et globale, l'alignement des réseaux peut être réalisé sur deux réseaux ou plus, ce qui correspond à l'alignement par paires (*cf.* Figure 2.21(a)) et à l'alignement multiple (*cf.* Figure 2.21(b)), respectivement.



(a) Alignement par paires.

(b) Alignement de réseaux multiples.

FIG. 2.21 : Alignement par paires et alignement de réseaux multiples. (a) Alignement par paires. (b) Alignement multiple. Reproduit de [Loch and Faisal, 2015].

Nous présentons ci-après la méthode PATHBLAST qui peut aligner deux réseaux et NetworkBLAST qui peut aligner plusieurs reseaux.

2.6.1 PathBlast

Kelley *et al.* [Kelley et al., 2004] ont proposé une première approche d'alignement de deux réseaux d'interactions de protéines. Comme BLAST produit un alignement de séquences des protéines, PathBlast produit un alignement des réseaux . PathBLAST prend en entrée deux réseaux d'interactions et un entier k, et produit en sortie un chemin d'alignement entre les deux réseaux, de longueur k, représentant le chemin avec le plus de similarités entre eux.

La méthode cherche des alignements avec des scores élevés entre les paires de chemins d'interactions protéiques. Les scores d'alignement des chemins sont calculés en fonction des similarités séquentielles et la qualité des interactions protéiques.

2.6.2 NetworkBLAST

Sharan *et al.* en 2005 [Sharan et al., 2005] ont proposé NetworkBLAST qui permet d'aligner jusqu'à trois réseaux simultanément en entrée. De plus, il peut détecter tous les structures conservés entre les réseaux. Plus précisément, NetworkBLAST fournit deux types de structure en sortie. Des chemins tels que PathBLAST, ainsi que, des clusters denses fortement associés aux régions protéiques. La Figure (*cf.* Figure 2.22) montre ces entrées et sorties.

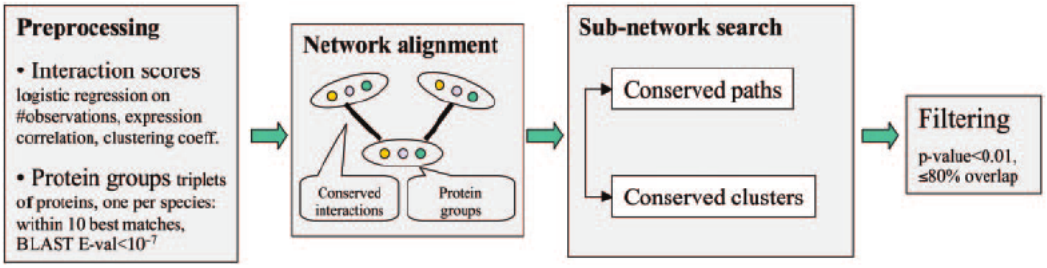


FIG. 2.22 : Illustration de l'alignement multiple de réseaux. Les protéines similaires entre les trois réseaux sont regroupées dans les noeuds du graphe, puis l'algorithme de recherche ressort les structures intéressantes. (Source [Sharan et al., 2005]).

2.7 Comparaison des réseaux biologiques hétérogènes

En biologie des systèmes, deux réseaux sont dits hétérogènes s'ils contiennent différents types d'informations, décrivant des aspects distincts de processus liés à la même entité biologique. La comparaison de ces deux réseaux hétérogènes consiste à chercher la même information biologique représentée de façon différente, contrairement à la comparaison de réseaux homogènes où l'information est représentée de la même manière dans les deux réseaux.

La comparaison de réseaux biologiques hétérogènes a débuté il y a plusieurs années[Ogata et al., 2000, Zheng et al., 2003, Rison et al., 2002, Lee et al., 2004, Durek and Walther, 2008, Huthmacher et al., 2008, Fertin et al., 2012, Babou, 2012, Zaharia et al., 2019, Akihiro Nakaya and Kanehisa, 2001, Boyer et al., 2005, Bordron et al., 2011] et reposait sur deux catégories de méthodes pour comparer deux réseaux hétérogènes d'une même espèce. La première catégorie contient des méthodes qui ont été proposées pour résoudre un problème très spécifique, alors que les méthodes de la deuxième catégorie sont générales et plus facilement adaptables à différents types de données biologiques.

La plupart des méthodes de la première catégorie consistent à chercher une chaine de réactions dans les réseaux métaboliques, en étudiant les interactions entre les protéines catalysant ces réactions [Durek and Walther, 2008, Huthmacher et al., 2008], la proximité génomique [Boyer et al., 2005, Lee et al., 2004, Ogata et al., 2000, Pál and Hurst, 2004, Rison et al., 2002, Zheng et al., 2003] et la co-expression des gènes [Ihmels et al., 2004, Kharchenko et al., 2004]. Nous présentons dans cette section quelques méthodes manuelles (ou cas par cas) pour la comparaison des réseaux hétérogènes puis ensuite des méthodes génériques pour la comparaison des réseaux hétérogènes.

2.7.1 Travaux préliminaires

Pour la résolution de problèmes spécifiques ("première catégorie"), nous présentons les méthodes ci-après.

2.7.1.1 Clusters de gènes corrélés

Ogata et al. [Ogata et al., 2000] ont proposé un algorithme heuristique de comparaison de graphes pour l'extraction des clusters d'enzymes fonctionnellement similaires. Cette méthode permet d'obtenir un ensemble d'enzymes catalysant des réactions successives dans une voie métabolique telle que les enzymes sont codées par des gènes voisins sur le chromosome.

Dans cette approche comparative, l'ordre des gènes sur le chromosome et les voies métaboliques sont modélisés par des graphes non-orientés. L'ordre des gènes sur le chromosome est représenté par un graphe non-orienté $G_1 = (V_1, E_1)$ dont les sommets représente les gènes. Si l'organisme étudié possède plusieurs chromosomes, alors G_1 a plusieurs composantes connexes. Une voie métabolique est représentée par un graphe non-orienté $G_2 = (V_2, E_2)$ dont les sommets représentent des enzymes. Deux sommets sont reliés par une arête si les enzymes qu'ils représentent sont impliquées dans des réactions partageant le même métabolite. Le mapping entre les deux graphes G_1 et G_2 est donné par une fonction de correspondance "plusieurs à plusieurs" basée sur les numéros EC entre V_1 et V_2. La correspondance est "plusieurs à plusieurs" car une enzyme donnée peut catalyser plusieurs réactions et une réaction donnée peut impliquer plusieurs enzymes.

Deux paramètres de *gap* γ_1 et γ_2 sont définis, représentant le nombre de gènes et d'enzymes qui peuvent être omis en G_1 et G_2, respectivement. Initialement, chaque paire de sommets correspondants dans V_1 et V_2 forme un cluster. Deux clusters C_i et C_j sont fusionnés s'il existe un chemin le plus court dans G_1 et G_2 entre un sommet de C_i et un sommet de C_j, de sorte que la longueur du chemin soit au plus égale à $\gamma_1 + 1$ dans G_1 et $\gamma_2 + 1$ dans G_2, respectivement. Les clusters sont fusionnés selon cette procédure jusqu'à ce qu'il n'y ait plus de clusters à fusionner.

[Akihiro Nakaya and Kanehisa, 2001] ont amélioré cet algorithme à fin de traiter des graphes multiples avec des gènes ou des produits de gènes comme sommets. La correspondance entre les sommets de deux graphes différents est établie à l'aide d'hyper-arêtes. Un identifiant de groupes de gènes corrélés est défini. Un groupe de gènes corrélés est un ensemble de sommets correspondants dans les graphes d'entrée. Dans la Figure 2.23, C_1 et C_2 sont deux groupes de gènes corrélés dans les graphiques d'entrée G_1, G_2 et G_3.

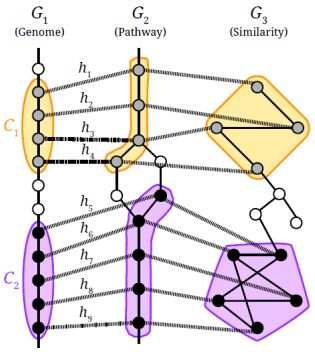


FIG. 2.23 : Clusters de gènes corrélés. C_1 (en jaune) et C_2 (en violet) sont deux clusters de gènes corrélés, reliés par des hyper-arêtes $h_1, ..., h_4 et h_5, ..., h_9$, respectivement [Akihiro Nakaya and Kanehisa, 2001]. (Source [Zaharia, 2018]).

2.7.1.2 Prédiction d'opéron

Observant que les enzymes codées par des gènes appartenant à un opéron ont tendance à catalyser des réactions successives, Zheng *et al.* [Zheng et al., 2003] ont développé une méthode de prédiction des opérons à partir de données métaboliques et génomiques.

Comme pour la méthode précédente (voir la section ci-dessus), les voies métaboliques et l'ordre

des gènes sur le chromosome sont représentés sous forme de graphes non-orientés. La correspondance entre les gènes et les enzymes est basée sur les numéros *EC*.

L'algorithme de prédiction des opérons est un processus en trois étapes :

- L'étape de mise en correspondance utilise une version modifiée de *breadth-first search* (BFS) dans laquelle chaque sommet du graphe représentant une voie métabolique est, à son tour, le sommet de départ dans l'arbre. Pour chaque sommet de départ, un arbre résultant de BFS est construite. Après la construction de cette arbre on vérifie si des sommets de l'arbre se trouvent dans une même région du chromosome. Comme cette méthode vise à prédire les opérons et non à identifier des clusters de gènes corrélés (voir la section ci-dessus), BFS fonctionne jusqu'à une profondeur prédéterminée (mais configurable) de 3, ce qui signifie que seules les réactions jusqu'à trois pas du sommet de la racine sont visitées. Par exemple, BFS est exécuté avec une profondeur de 2 en partant du sommet noir dans la Figure 2.24 et à la fin de cette étape, les opérons putatives sont identifiés.
- L'étape d'élagage a pour but d'augmenter la spécificité de l'algorithme. Elle consiste à éliminer les gènes aux extrémités des opérons identifiés lors de l'étape précédente s'ils sont séparés des autres gènes du groupe par au moins deux autres gènes.
- L'étape de clustering a lieu à la fin, une fois que les étapes de mise en correspondance et d'élagage ont été effectuées pour chaque sommet de chaque voie métabolique de l'espèce étudiée. L'étape de clustering consiste à fusionner les clusters se chevauchant après les étapes de mise en correspondance et d'élagage.

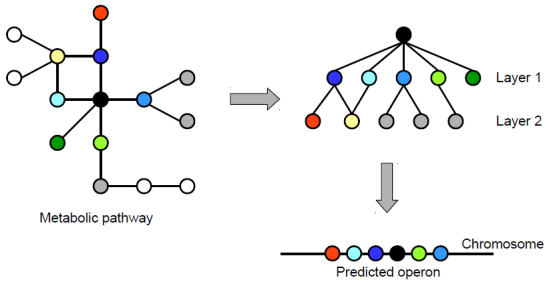


FIG. 2.24 : Représentation graphique de *Breadth-First Search* (BFS). Ici, la BFS commence avec le sommet en noir dans la voie métabolique. Dans cet exemple, BFS est lancé pour une profondeur de 2. Le premier niveau contient les voisins directs du sommet noir. Le deuxième niveau contient les voisins directs des sommets du premier niveau [Zheng et al., 2003]. (Source [Zaharia, 2018]).

2.7.1.3 Modules évolutifs

[Spirin et al., 2006] ont intégré les réseaux métaboliques et les associations génomiques à fin de révéler les modules évolutifs. Les modules évolutifs sont définis comme des régions du réseau métabolique constituées de réactions hautement connectées qui sont également hautement associées d'un point de vue génomique. On dit que deux gènes sont associés si, dans des organismes différents, leurs voisinages sont conservés, s'ils présentent une co-occurrence, et/ou s'ils peuvent être trouvés reliés entre eux.

Le réseau métabolique et génomique intégré est un graphe non-orienté avec des réactions pour les sommets, reliés par deux types d'arêtes représentant respectivement les associations métaboliques et génomiques. respectivement. Une arête métabolique connecte deux réactions si elles partagent un métabolite. Les métabolites omniprésents (tels que l'ATP, le phosphate, H+, etc.) sont exclus à fin d'éviter une surconnectivité des réactions. Deux réactions sont connectées par un arête génomique si elles sont catalysées par des enzymes ou des sous-unités d'enzymes codées par des gènes associés.

2.7.2 Frameworks pour la comparaison des réseaux hétérogènes

Pour la résolution de problèmes généraux (“seconde catégorie”), nous présentons les méthodes ci-après.

2.7.2.1 Recherche des motifs dans un graphe correspondance

Boyer *et al.* [Boyer et al., 2005] ont conçu un cadre pour l'extraction des motifs à partir d'un graphe non-orienté appelé multigraphe de correspondance représentant les réseaux en entrée et les relations entre eux.

Dans le multigraphe de correspondance les sommets sont reliés par différents types d'arêtes. Le type d'arêtes est défini en fonction d'une relation de correspondance. Par exemple, les sommets peuvent être des réactions et deux types différents d'arêtes entre les sommets peuvent décrire les réactions en interaction dans une voie métabolique et celles catalysées par les produits des gènes voisins. Un multigraphe de correspondance est similaire au réseau métabolique-génomique intégré proposée par [Spirin et al., 2006]. En changeant la relation de correspondance, le multigraphe peut accueillir différents types de données. Par exemple, il peut être utilisé pour représenter les protéines en interaction en fonction de l'ordre des gènes sur le chromosome. Dans le multigraphe, les composantes connexes communes sont des sous-graphes maximaux tels que pour tous les types d'arêtes dans le multigraphe deux sommets quelconques sont reliés par des chemins constitués d'arêtes d'un type donné.

2.7.2.2 $SIPPER$: une méthode d'intégration et d'analyse de données omiques hétérogènes

Bordron *et al.* [Bordron et al., 2011] ont étudié deux hypothèses biologiques à partir desquelles ils ont appliqué une méthode appelée SIPPER, à fin d'étudier quelles informations biologiques chaque hypothèse vérifie et discrimine. La première application de SIPPER étudie l'hypothèse que des réactions métaboliques qui s'enchaînent sont catalysées, via des enzymes, par des gènes proches sur le génome. La seconde application étudie l'hypothèse que des réactions métaboliques qui s'enchaînent sont catalysées, via des enzymes, par des gènes coexprimés. Bordron $et\ al.$ [Bordron et al., 2011] ont présenté $SIPPER$, une méthode qui était appliquée sur le réseau génomique et métabolique intégré d'Escherichia coli. $SIPPER$ retourne les k plus courts chemins entre deux réactions.

Le réseau intégré est un graphe orienté pondéré où chaque sommet est étiqueté avec une paire gène-réaction. Les poids des arcs dans le réseau intégré représentent la distance entre les gènes au sein du génome. Les poids des arcs sont utilisés pour calculer la longueur des chemins. Le chemin intégré le plus court entre deux réactions est appelé un 1-SIP. $SIPPER$ utilise un algorithme heuristique pour calculer les k chemins les plus courts entre une réaction source et une réaction destination. Ceci qui permet d'obtenir un sous-graphe du réseau intégré appelé k-SIP. Deux hypothèses biologiques ont été étudiées à partir desquelles la méthode $SIPPER$ est appliquée en tant que méthode exploratoire, à fin d'étudier quelles informations biologiques. La première application de $SIPPER$ étudie l'hypothèse que des réactions métaboliques qui s'enchaînent sont catalysées, via des enzymes, par des gènes proches sur le génome. La seconde application étudie l'hypothèse que des réactions métaboliques qui s'enchaînent sont catalysées, via des enzymes, par des gènes coexprimés. Ils ont généré deux instances de G_{int}, nommées G_{col} et G_{coexp}, puis calculer, pour chacune de ces instances et pour k de 1 à 10, l'ensemble des $k-SIPs$ entre singletons de réactions. Ils ont ensuite analysé à la fois d'un point de vue génomique.

2.7.2.3 Plus long chemin DG-consistant

Des études récentes ont recherché des motifs métaboliques et génomiques d'une espèce donnée dans deux réseaux biologiques hétérogènes modélisés respectivement par un graphe orienté D et un graphe non-orienté G'. Par exemple, D peut représenter un réseau métabolique et G' peut représenter

l'ordre des gènes sur le chromosome, ou un réseau d'interaction protéine-protéine [Fertin et al., 2012, Babou, 2012, Zaharia et al., 2019]. Babou *et al.* [Babou, 2012] ont proposé un framework pour la comparaison sur la base duquel ils ont introduit un problème d'optimisation combinatoire appelé "One-to-One SkewGraM". Ce problème est issu du modèle ci-après.

- **Modèle**
  Une réaction métabolique non spontanée est catalysée par une ou plusieurs enzymes. Une enzyme donnée peut être codée par un ou plusieurs gènes. Les voies métaboliques et le contexte génomique sont considérés comme des réseaux de réactions et de gènes, respectivement. La relation entre les voies métaboliques et les gènes qui les codent est représentée par un modèle classique impliquant deux graphes et une fonction de correspondance :
  - Les génomes (considérés comme des réseaux de gènes) sont représentés par des graphes non-orienté dont les sommets sont des gènes codant des protéines (Voir graphe G' dans la Figure 2.25). Deux gènes codant des protéines sont reliés par une arête s'ils sont voisins sur le même brin du même chromosome. Par exemple, les gènes A et B dans G' sont voisins.
  - Les voies métaboliques sont représentées sous forme de graphes orientés dont les sommets représentent les réactions et un arc entre deux réactions signifie que ces deux réactions partageant le métabolite (Voir graphe D dans la Figure 2.25).
  - Pour une espèce donnée S, la relation entre une de ses voies métaboliques et son génome prend la forme d'une fonction de correspondance associant les gènes à la voie métabolique. Son génome se présente sous la forme d'une fonction de correspondance associant les gènes aux réactions : pour toute réaction r donnée, la fonction de correspondance renvoie l'ensemble des gènes de l'espèce S qui codent les enzymes catalysant la réaction r (par exemple, dans la Figure 2.25, le gène A dans G' code l'enzyme catalysant la réaction r_5 dans D). Ces informations peuvent être trouvées dans des bases de données telles que KEGG qui contient pour une espèce donnée des informations sur ses voies métaboliques, les réactions que l'espèce effectue, et les gènes associés à ces réactions (voire la section 2.7.3).

Ce framework nécessite deux simplifications, car il prend en entrée un graphe orienté acyclique (DAG) D et un graphe non-orienté G qui doit être construit à partir des informations de D et G' et une fonction de correspondance qui associe à un chaque sommet dans D un ensemble de sommets dans G'. Cette fonction de correspondance est utilisée pour construire G en traduisant chaque arête dans G'. L'algorithme compare indirectement D et G' en comparant D avec le graphe supplémentaire G. Le graphe G est construit comme suit : Nous avons deux graphes D (orienté) et G' (non-orienté). A partir des deux graphes on construit un graphe non-orienté G tel que $V(G) = V(D)$. On mets une arête dans G entre deux sommets $i, j \in V(G)$ si seulement si les réactions i et j sont catalysés par des enzymes produites par des gènes adjacents dans G'.

L'objectif est de chercher une chaine de réactions qui sont catalysées par le produit des gènes voisins. Ces chemins peuvent être exploités pour étudier la conservation des motifs métaboliques et génomiques entre différentes espèces. Nous présentons ci-après la formulation du problème.

- **Formulation du problème**
  Étant donné une voie métabolique et le contexte génomique pour la même espèce, les motifs métaboliques et génomiques capturés par une méthode de recherche de chemins sont des chaînes maximales de réactions catalysées par les produits de gènes voisins.
  Le problème de comparaison est un problème de maximisation appelé Skew SubGraph Mining (abrégé SkewGraM). Le problème SkewGraM est défini comme suit.

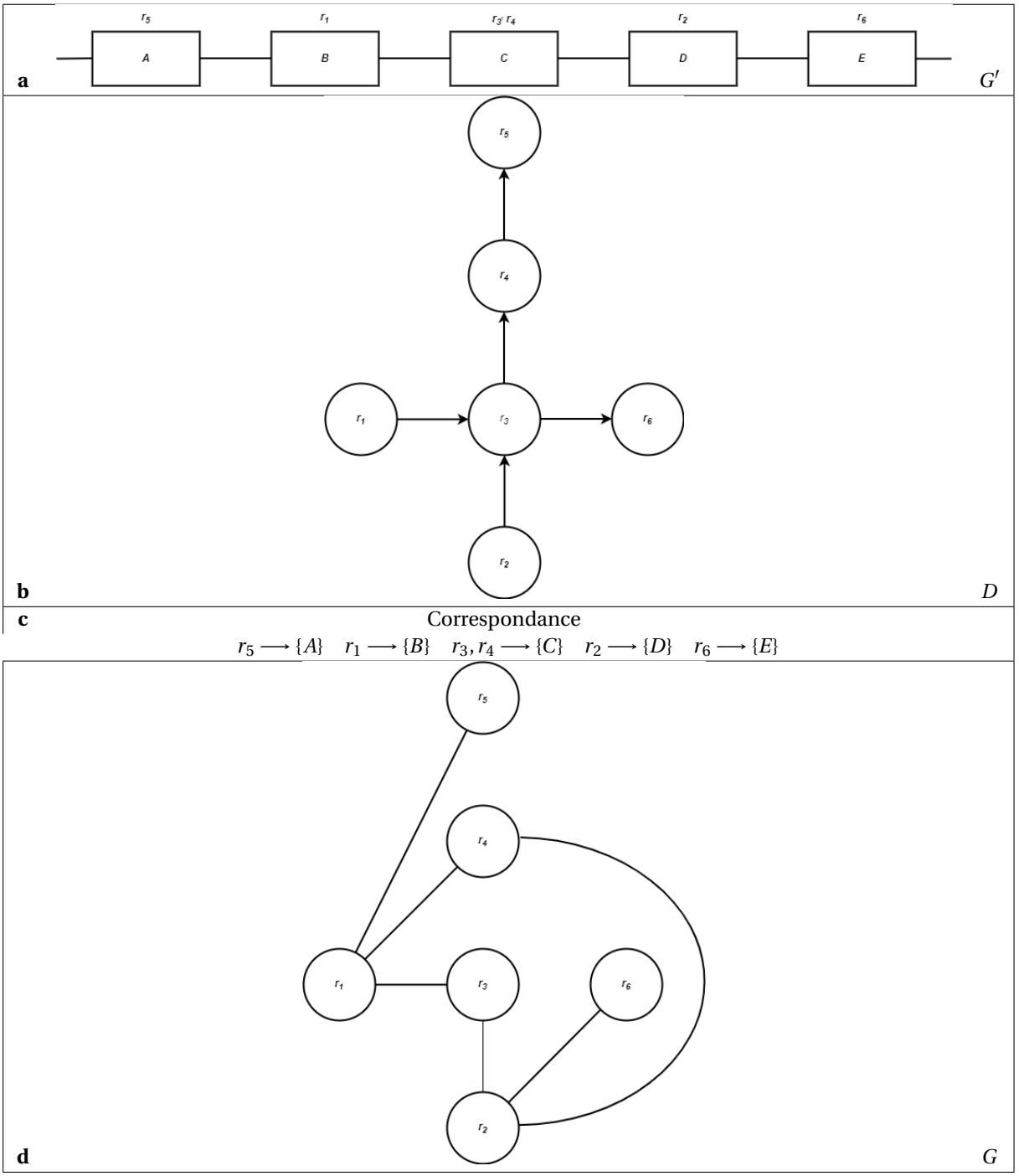


FIG. 2.25 : Illustration du modèle utilisé pour représenter les voies métaboliques et le contexte génomique. (a) Le graphe non-orienté G' représente l'ordre des gènes d'une espèce donnée. Les réactions que les produits des gènes catalysent sont indiquées au-dessus de chaque gène. (b) Le graphe orienté D représente une voie métabolique de la même espèce que dans (a). (c) La correspondance entre les réactions dans D et les gènes dans G'. (d) G est un graphe non-orienté construit sur le même ensemble de sommets que D en utilisant la correspondance entre les réactions et les gènes.

SkewGraM

Instance : Un graphe orienté D, un graphe non-orienté G.

Solution : Un chemin P de D, tel que $G[V(P)]$ est connexe.

Mesure : La longueur du chemin P de D, tel que $G[V(P)]$ est connexe.

En raison des motivations algorithmiques et biologiques, un cas particulier du problème du problème SkewGraM appelé One-To-One SkewGraM où D est acyclique (DAG) a été étudié dans la littérature. Ce problème est formulé comme suit.

One-To-One SkewGraM

Instance : Un DAG D, un graphe non-orienté G.

Solution : Un chemin P de D, tel que $G[V(P)]$ est connexe.

Mesure : La longueur du chemin P de D, tel que $G[V(P)]$ est connexe.

Nous présenterons des méthodes pour résoudre le problème One-To-One SkewGraM dans le chapitre 3.
La solution pour le problème One-To-One SkewGraM est donc le plus long chemin dans le graphe orienté D induisant un sous-graphe connecté dans le graphe non-orienté G. La grande majorité des voies métaboliques présentent cependant des cycles (par exemple, des réactions réversibles). La prise en compte des cycles exige que les sommets peuvent apparaitre plusieurs fois dans la solution.
Pour traiter le cas général (quand D contient des cycles), récemment, Zaharia *et al.* [Zaharia et al., 2019] ont développé une méthode qui prend en entrée un graphe orienté D qui peut contenir des cycles et le graphe supplémentaire G et donne comme sortie un *trail* des réactions qui sont catalysées par le produit des gènes voisins. Un *trail* est un chemin élémentaire, un sommet peut apparaître plusieurs fois mais un arc est présent au plus une fois.
Le choix d'extraire *trails* est motivée par trois aspects. D'abord, le *trail* est le seul motif capable de capturer les cycles des voies métaboliques. Ensuite, il s'agit de *trails* dans le graphe orienté, et l'orientation est préservée. Enfin, les *trails* garantissent que les motifs extraits correspondent à des routes métaboliques réelles, contrairement aux méthodes d'extraction des sous-graphes.
Nous présenterons des méthodes pour résoudre le problème SkewGraM dans le chapitre 4.

À fin de tester nos méthodes sur des données biologiques réelles, nous allons extraire des informations d'une base de données portant sur les voies métaboliques identifiées de différentes espèces. Nous présentons cette base de données dans la 2.7.3.

2.7.3 Les bases de données biologiques

Il existe différents types des bases de données qui s'alimentent les unes des autres (*e.g.* génomiques, métaboliques, protéiques, ...). Dans cette thèse, nous étudions les données issues d'une base de données métaboliques [Wittig and De Beuckelaer, 2001]. On distingue deux types de bases de données répertoriant les voies métaboliques, celles qui les répertorient de manière individuelle et celles qui répertorient des reconstructions du métabolisme où l'ensemble des voies sont inter-connectées. La base des données SMPDB [Frolkis et al., 2009] fait partie des bases de données répertoriant les voies métaboliques

individuelles qui ont des utilités multiples, notamment pour la visualisation et l'analyse des données expérimentales. En ce qui concerne les bases de données répertoriant les reconstructions du métabolisme, nous citons comme exemples MetExplore qui est maintenu depuis 2010 et KEGG (Kyoto Encyclopedia of Genes and Genomes) [Cottret et al., 2018, Ogata et al., 1999] et . Cette dernière est majoritairement mise à profit dans le cadre d'analyses de prédiction de phénotype et/ou de réponses aux perturbations biologiques par des analyses de flux. Dans cette thèse, nous utilisons la base de données KEEG.

KEGG est un ensemble de bases de données et de ressources permettant d'étudier les fonctions de haut niveau et les utilités des systèmes biologiques [Kanehisa and Goto, 2000, Kanehisa et al., 2013]. Le projet de base de données KEGG a débuté en 1995 à l'Institut de recherche chimique de l'Université de Kyoto, à la recherche d'une représentation informatisée des liens entre les informations génomiques et les fonctions systémiques de haut niveau de la cellule, de l'organisme et de l'écosystème (*cf.* Figure 2.26).

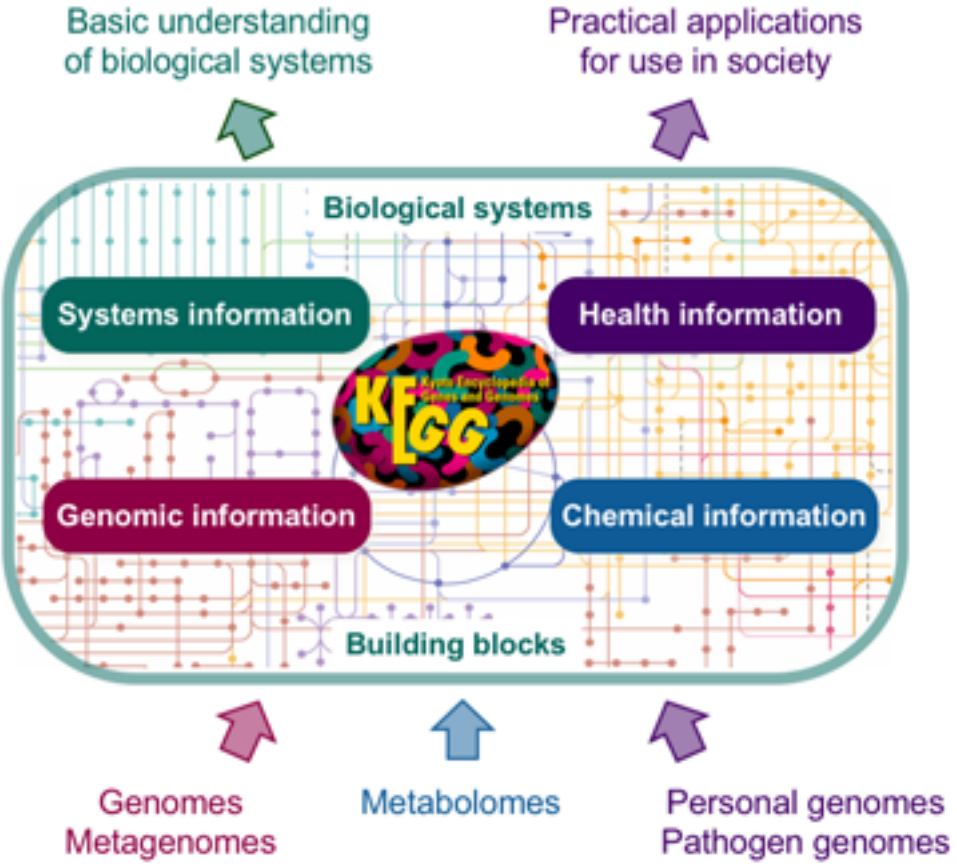


FIG. 2.26 : Vue d'ensemble de l'information intégrée dans la base de données KEGG. (via KEGG).

La dernière version de la base de données KEGG (version 100.0, 1er octobre 2021) comprend 18 bases de données principales qui contiennent de larges informations génomiques et moléculaires de plus de 7000 espèces (660 eucaryotes, 6536 bactéries et 357 archées). Ces bases de données sont classées en quatre grandes catégories : informations sur les systèmes, informations génomiques, informations chimiques et informations sur la santé (*cf.* Figure 2.27).

En décembre 2017, la base de données KEGG NETWORK a été publiée ainsi que la base de données associée de KEGG VARIANT. Ces deux bases de données sont spécifiques à l'espèce humaine et font

partie de la catégorie des informations sur la santé (*cf.* Figure 2.26).

Bien qu'il n'est considéré dans cette base de données que les variations pertinentes pour les maladies et les médicaments humains, la méthodologie utilisée dans KEGG NETWORK peut être appliquée à toutes les variations de n'importe quelle espèce. Une description plus complète de la base de données KEGG peut être trouvée dans [Kanehisa et al., 2017].

Category	Database name	Content	Release
Systems Information	KEGG PATHWAY	KEGG pathway maps	1995
	KEGG BRITE	BRITE functional hierarchies and tables	2005
	KEGG MODULE	KEGG modules	2006
Genomic Information	KEGG ORTHOLOGY (KO)	KO groups for functional orthologs	2002
	KEGG GENOME	KEGG organisms (complete genomes)	2000
	KEGG GENES	Genes and proteins	1995
	KEGG SSDB	Sequence similarity among GENES entries	2001
Chemical Information	KEGG COMPOUND	Metabolites and other small molecules	1995
	KEGG GLYCAN	Glycans	2003
	KEGG REACTION / RCLASS	Biochemical reactions and reaction class	1998/2010
	KEGG ENZYME	Enzyme nomenclature	1995
Health Information	KEGG NETWORK / VARIANT	Disease-related network and gene variants	2017
	KEGG DISEASE	Human diseases	2008
	KEGG DRUG / DGROUP	Drugs and drug groups	2005/2014
	KEGG ENVIRON	Crude drugs and health-related substances	2010
	JAPIC	Japanese drug labels	2007
	DailyMed	FDA drug labels (links only)	2012

FIG. 2.27 : KEGG se compose de dix-huit bases de données réparties en quatre catégories, qui sont toutes gérées manuellement, à l'exception de la SSDB générée par calcul. Les bases de données de la catégorie des informations chimiques sont collectivement appelées KEGG LIGAND. Les bases de données de la catégorie des informations sanitaires, ainsi que deux bases de données externes, les étiquettes de médicaments japonaises obtenues à partir de la base de données JAPIC (http ://www.japic.or.jp) et les étiquettes de médicaments de la FDA liées à la base de données DailyMed (https : //dailymed.nlm.nih.gov), sont collectivement appelées KEGG MEDICUS. (via KEGG).)

La base de données KEGG PATHWAY est une collection de diagrammes graphiques, généralement connus sous le nom de cartes de voies, qui représentent les réseaux d'interactions et de réactions moléculaires au sein d'une cellule au cours de processus biochimiques spécifiques, qui aboutissent généralement à un produit ou à un changement dans la cellule. KEGG PATHWAY contient environ 548 voies de référence (version 100.0, 1er octobre 2021), qui sont dessinées manuellement et continuellement mises à jour en fonction des preuves biochimiques ; elles sont classées selon la classification hiérarchique de la Figure 2.28. En plus des cartes de référence, KEGG PATHWAY contient plus de 800 000 voies spécifiques aux organismes déduites par cartographie automatique basée sur les événements d'orthologie existants entre les espèces.

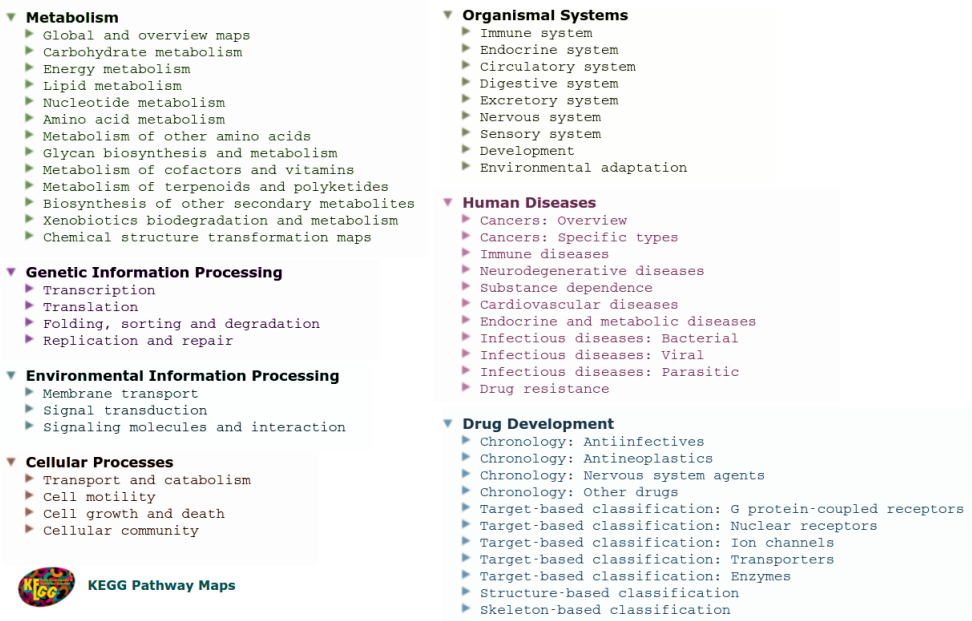


FIG. 2.28 : Classification hiérarchique des voies KEGG : catégories principales et secondaires. (via KEGG).

Chaque voie dans KEGG est identifiée par un numéro à 5 chiffres (le nom de l'entrée ou le numéro d'accès) précédé d'un code de 2 à 4 lettres qui indique l'organisme ou les bases de données. Quelques exemples d'identifiants valides sont map03060 (voie d'exportation des protéines de référence, Figure 2.29), mmu03060 (voie d'exportation des protéines de Mus Musculus) ou hsa03060 (voie d'exportation des protéines de Homo Sapiens).

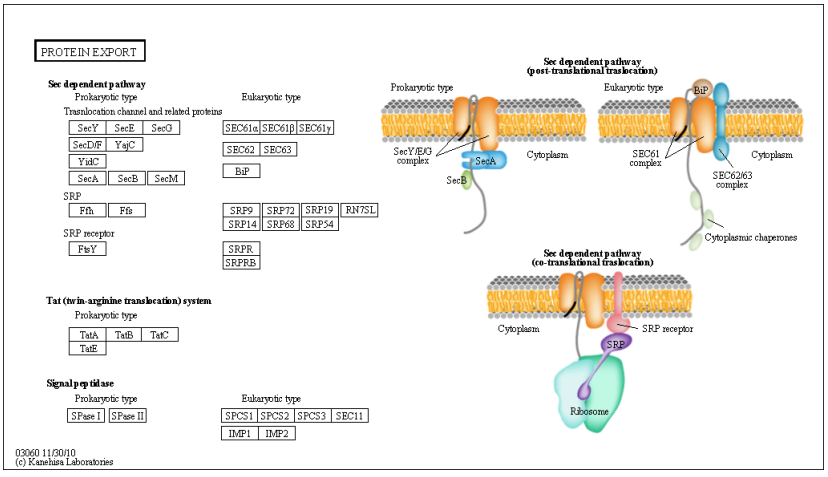


FIG. 2.29 : Diagramme KEGG pour la voie d'exportation des protéines de référence (KEGG ID map03060). Classification hiérarchique des voies KEGG : catégories principales et secondaires. (via KEGG).

Les voies sont dessinées manuellement à l'aide d'un logiciel interne appelé KegSketch, en utilisant différentes ressources graphiques pour visualiser les informations. Par exemple, les boîtes représentent les produits des gènes, principalement des protéines mais aussi de l'ARN, tandis que les cercles représentent d'autres molécules telles que des composés chimiques (*cf.* Figure 2.30a). Les interactions entre

biomolécules ou autres voies sont dessinées à l'aide de différentes flèches (*cf.* Figure 2.30b) ; et la combinaison de plusieurs formes peut être interprétée comme différents processus biochimiques ou interactions moléculaires (*cf.* Figure 2.30c). La coloration est également une autre ressource pour l'interprétation des diagrammes : en règle générale, les voies de référence ne sont pas colorées, tandis que les variations des voies pour la base de données KEGG ENZYME (EC) sont colorées en bleu, et les voies spécifiques à un organisme sont colorées en vert, où la coloration indique que l'entité biologique (c'est-à-dire le gène ou le composé) existe dans la base de données correspondante (figure 2.30d).

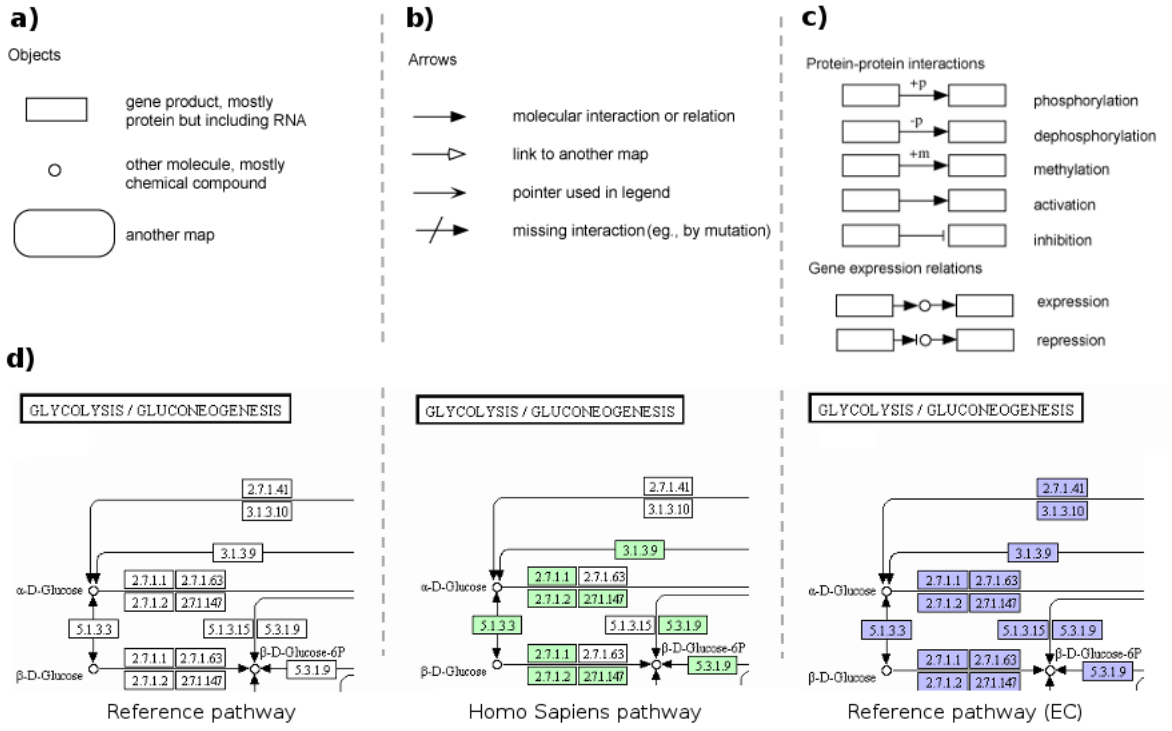


FIG. 2.30 : Quelques exemples des ressources graphiques utilisées pour le diagramme KEGG pour visualiser l'information. (via KEGG).

Enfin, il est intéressant de souligner l'existence d'un format d'échange pour les cartes de la voie KEGG connu sous le nom de KEGG Markup Language (KGML), qui contient des informations informatiques sur les objets graphiques et leurs relations dans la voie KEGG, c'est-à-dire les coordonnées des formes, les dimensions, les couleurs ou les liens vers les bases de données, entre autres (fragment de code 1).

```
<?xml version="1.0"?>
<!DOCTYPE pathway SYSTEM "http://www.kegg.jp/kegg/xml/KGML_v0.7.1_.dtd">
<pathway name="path:hsa00010" org="hsa" title="Glycolysis/Gluconeogenesis"
    image="http://www.kegg.jp/.../hsa00010.png" link="...">
<entry id="13" name="hsa:226 hsa:229 hsa:230" type="gene" reaction="rn:R01070" link="...">
<graphics type="rectangle" x="483" y="407" width="46" height="17" name="ALDOA, ALDA,
    GSD12,..." fgcolor="#000" bgcolor="..." />
</entry>
<entry id="37" name="hsa:217 hsa:219 hsa:223 hsa:224 hsa:501" type="gene"
    reaction="rn:R00710">
[...]
```

```
</entry>
[...]
<entry id="113" name="cpd:C00036" type="compound"
link="http://www.kegg.jp/dbget-bin/www_bget?C00036">
<graphics name="C00036" fgcolor="#000000" bgcolor="#FFFFFF" type="circle" x="146" y="736"
    width="8" height="8"/>
</entry>
[...]
<relation entry1="68" entry2="70" type="ECrel">
<subtype name="compound" value="86"/>
</relation>
[...]
<reaction id="47" name="rn:R00014" type="irreversible">
<substrate id="98" name="cpd:C00022"/>
<substrate id="136" name="cpd:C00068"/>
<product id="99" name="cpd:C05125"/>
</reaction>
[...]
</pathway>
```

Il est possible de comprendre la fonction de chaque composant biologique par la comparaison des séquences. Dans la section suivante, nous présentons des méthodes de la comparaison des séquences.

2.8 Conclusion

Dans ce chapitre, nous avons présenté des notions de base de biologie ainsi que l'état de l'art sur l'alignement des séquences et réseaux. Nous avons également présenté des approches utilisées pour comparer des réseaux biologiques hétérogènes. Nous proposerons également de nouvelles méthodes de comparaison de ces réseaux dans les chapitres 3, 4 et 5.

Chapitre 3

Le problème One-To-One SkewGraM

Contents

3.1 Introduction

Dans ce chapitre, nous allons proposer des méthodes exactes pour la résolution d'un problème de la littérature nommé One-To-One SkewGraM, qui consiste à calculer le plus long chemin de réactions dans un réseau métabolique, de sorte que ces réactions soient toutes catalysées par des produits de gènes voisins. Autrement dit, le plus long chemin P dans un graphe orienté $D = (V, A)$, tel que les sommets du chemin P induisent un sous-graphe connexe d'un graphe non-orienté $G = (V, E)$ (les deux graphes sont construits sur le même ensemble de sommets). Ces chemins sont utilisés par les biologistes pour comparer les espèces. Ce problème est issu d'un modèle de comparaison de réseaux hétérogènes qui a été développé pour tenir compte de l'orientation des arcs dans les réseaux métaboliques. Ce modèle a des applications dans l'analyse des réseaux biologiques et des applications prévisibles dans l'analyse des réseaux sociaux, d'information et de communication.

Nous commencerons par "AlgoBB" de la littérature qui est un algorithme de branch-and-bound ayant été publié [Babou, 2012] et qui prend comme arguments un graphe orienté acyclique $D = (V, A)$ (un réseau métabolique), un graphe non-orienté $G = (V, E)$ (un graphe codant la proximité des gènes),

et un arc (i, j) de D. La sortie de cet algorithme est le plus long chemin (D, G)-Consistant (c'est-à-dire un chemin satisfaisant la contrainte de proximité entre les gènes) passant par l'arc (i, j). Pour améliorer cet algorithme, nous allons proposer de nouvelles règles de dominance et une nouvelle politique de sélection pour les arcs qui nous permettent de résoudre le problème plus efficacement. Nous proposerons également deux formulations de programmation linéaire en nombres entiers (ILP) en adaptant et en combinant des formulations pour le problème du plus long chemin et le problème de l'arbre couvrant. Nous présenterons ensuite un modèle de programmation par contraintes qui s'avère plus efficace que les autres méthodes. Nous évaluons les performances de ces méthodes en termes de temps d'exécution et de qualité en les appliquant sur des graphes aléatoires et des données biologiques. Dans cette thèse, la recherche d'un chemin se concentre sur les voies métaboliques et le contexte génomique. Les méthodes sont cependant adaptables à d'autres types.

3.2 Modèle

Ce modèle a pour but d'effectuer des études multi-échelles en biologie des systèmes pour prédire des structures biologiquement significatives. Il a des applications immédiates dans l'analyse des réseaux biologiques et des applications prévisibles dans l'analyse des réseaux sociaux, d'information, et de communication.

Dans le contexte de la biologie des systèmes, Babou $et\ al.$ [Babou, 2012] ont comparé un réseau métabolique avec un réseau d'interaction proteine-proteine. Zaharia $et\ al.$ [Zaharia et al., 2019] ont étudié la relation entre le métabolisme et le génome en comparant le réseau métabolique avec le génome (réseau de gènes).

Dans cette thèse, nous nous intéressons à l'étude des relations entre le métabolisme et le génome. Nous allons comparer deux réseaux biologiques hétérogènes D et G'. Un réseau métabolique peut être représenté par un graphe orienté D dont les sommets représentent les réactions. Il existe un arc entre deux réactions r_i et r_j dans D si un produit de r_i est un substrat de r_j. Un génome est représenté par un graphe non-orienté G' dont les sommets représentent les gènes. Il existe une arête entre deux gènes g_1 et g_2 s'ils sont voisins et sur le même brin du même chromosome.

Comme nous l'avons vu dans la section 2.7.2.3, il est possible de construire un graphe G, à partir de G', dont les sommets sont des réactions (les même que dans D) et dont les arcs reflètent le voisinage des gènes par rapport aux réactions que les produits de ces gènes catalysent.

3.3 Définition du problème

Sur la base du modèle présenté dans la section précédente, Fertin et *al* [Fertin et al., 2012] ont introduit un problème d'optimisation combinatoire appelé "One-to-One SkewGraM." Ce problème est défini comme suit. Soit $D = (V, A)$ un graphe orienté acyclique et $G = (V, E)$ un graphe non-orienté construits sur le même ensemble de sommets $V = \{1, 2, \ldots, n\}$. Un chemin (D, G)-consistant P est un chemin (dirigé) dans D tel que les sommets de P induisent un sous-graphe connexe dans G. Le problème One-to-One SkewGraM consiste à calculer le plus long chemin (D, G)-consistant (*cf.* Figure 3.1). Dans le contexte décrit précédemment (où D modélise un réseau métabolique et G reflète le voisinage des gènes par rapport aux réactions que les produits de ces gènes catalysent), il s'agit de se concentrer sur la détection de chaînes de réactions catalysées par des produits de gènes voisins, où la notion de voisinage peut être modulée en autorisant que certaines réactions et/ou gènes soient omis d'une distance δ_D et/ou δ_G.
Babou *et al.* [Babou, 2012] ont étudié la complexité du problème en prouvant la $\mathcal{NP}$-difficulté au sens fort de ce problème dans le cas général. Ils ont également étudié la complexité paramétrée du problème en identifiant des cas traitables à paramètres fixes et $\mathcal{W}[1]$-durs. Malheureusement, cette étude n'est pas suffisamment poussée sur le plan algorithmique. Pour résoudre le cas général, ils ont déve-

loppé des méthodes heuristiques et exactes par séparation et évaluation (*branch-and-bound*). La méthode branch-and-bound prend en arguments un graphe orienté acyclique $D = (V, A)$, un graphe non-orienté $G = (V, E)$, et un arc $(i, j) \in A$. Cet algorithme, appelé AlgoBB, calcule le plus long chemin (D, G)-consistant passant par un arc (i, j). Pour trouver la solution optimale, AlgoBB est appelé $|A| = m$ fois (un nouvel arc est considéré à chaque itération). Cependant, cette approche est inefficace, même pour de petites instances.

Nous proposons d'abord une méthode de branch-and-bound basée sur la méthode AlgoBB que nous appelons "AlgoBB++". Nous proposons ensuite deux formulations de programmation linéaire en nombres entiers (ILP) en adaptant et en combinant des formulations pour le problème du plus long chemin et le problème de l'arbre couvrant. Nous proposons enfin une méthode de programmation par contraintes pour résoudre ce problème.

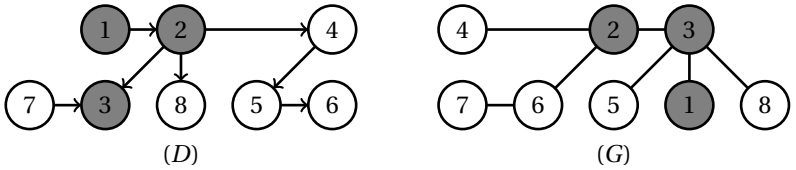


FIG. 3.1 : Exemple. Le plus long chemin (D, G)-consistant est $1 \to 2 \to 3$. En revanche, le plus long chemin dans D est $1 \to 2 \to 4 \to 5 \to 6$. Cependant, ce chemin n'est pas (D, G)-consistant parce que le sous-graphe de G induit par l'ensemble de sommets $\{1, 2, 4, 5, 6\}$ n'est pas connexe.

3.4 Algorithme de Branch-and-Bound AlgoBB++

AlgoBB [Babou, 2012] est un algorithme branch-and-bound qui prend comme arguments un graphe orienté acyclique $D = (V, A)$, un graphe non-orienté $G = (V, E)$, et un arc (i, j) dans D. AlgoBB est composé de deux parties, dont la première, appelée $CoverSet$, calcule l'ensemble couvrant de (i, j), c'est-à-dire le sous-ensemble maximal de sommets qui pourraient étendre l'arc (i, j) à un chemin P de D tel que les sommets de P induisent un sous-graphe connexe dans G. L'ensemble couvrant d'un arc donné est unique. Les graphes D et G, et donc l'espace de recherche à explorer, sont réduits. La deuxième partie de l'AlgoBB est une procédure par séparation et évaluation qui construit un arbre TS de solutions (D, G)-consistantes comme suit : La racine de l'arbre TS est associée à l'arc (i, j). À chaque niveau de l'arbre, un arc est ajouté à une extrémité du chemin actuel. Chaque noeud s de TS est alors associé à un chemin $P(s)$ qui prolonge l'arc (i, j). Les feuilles sont associées à des chemins (D, G)-consistants qui contiennent (i, j).

Séparation. Nous explorons un noeud s, associé à un chemin $P(s)$ dans D du sommet v_l au sommet v_m, comme suit : Pour chaque prédécesseur (successeur) $v_k \in V$ de v_l (v_m), nous ajoutons à TS un noeud fils de s associé au chemin obtenu en ajoutant v_k au début (à la fin) de $P(s)$. Pour tous les noeuds s de TS, $value(P(s))$ indique la longueur du plus long chemin dans le sous-graphe induit par les sommets de l'ensemble couvrant du chemin $P(s)$ dans D et G. Nous utilisons la fonction $BBvalue$ pour évaluer les noeuds de TS. $BBvalue$ est définie comme suit :

- Si s n'a pas encore été exploré, alors $BBvalue(s) = value(P(s))$.
- Si s a déjà été exploré, alors nous distinguons deux cas.
  - Le chemin $P(s)$ est (D, G)-consistant. Dans ce cas, $BBvalue(s)$ est égale à la longueur du chemin $P(s)$.
  - Le chemin $P(s)$ n'est pas (D, G)-consistant. Dans ce cas, $BBvalue(s) = 0$.

Stratégie de sélection de noeud. Soit $\{s_1, s_2, s_3, \ldots, s_k\}$ l'ensemble des noeuds à explorer. Nous choisissons le noeud s tel que $s = \text{argmax}_{1 \le i \le k} BBvalue(s_i)$. S'il existe plusieurs noeuds s_i dont $BBvalue(s_i)$ est

maximal, nous choisissons l'un d'eux arbitrairement.

Élagage. Soit s_{max} un noeud de TS satisfaisant les conditions suivantes : (i) s_{max} a déjà été exploré, et (ii) pour tous les noeuds s de TS, si s a déjà été exploré, alors $BBvalue(s_{max}) \geq BBvalue(s)$. Nous supprimons de TS toutes les feuilles s telles que $BBvalue(s) \leq BBvalue(s_{max})$. Cette suppression est appliquée récursivement sur les noeuds (sauf s_{max}), qui deviennent des feuilles après la suppression de leurs enfants.

La procédure de séparation et d'évaluation est définie comme suit :

1. Construire et évaluer la racine de l'arbre TS associée à l'arc (i, j).
2. $tant \quad que$ (TS contient un noeud s à explorer) $faire$
   (a) sélectionner un noeud s de TS;
   (b) séparer TS selon le noeud s;
   (c) brancher sur s et évaluer ses enfants.
3. Soit s_{max} une feuille vérifiant $BBvalue(s_{max}) \geq BBvalue(s)$ pour toute feuille s de TS. Soit $P_{max} = P(s_{max})$.
4. Si P_{max} est (D, G)-consistant, alors P_{max} est le plus long chemin (D, G)-consistant passant par (i, j). Sinon, aucun chemin (D, G)-consistant ne passe par (i, j).

La sortie de l'algorithme est le plus long chemin (D, G)-consistant passant par l'arc (i, j). Pour résoudre le problème One-to-One SkewGraM défini dans la section précédente, les auteurs appliquent simplement l'algorithme AlgoBB à tous les arcs de D et gardent le plus long chemin (D, G)-consistant parmi les chemins calculés [Fertin et al., 2012, Babou, 2012].

L'objectif de l'algorithme de branch-and-bound, AlgoBB++, est de trouver le plus long chemin (D, G)-consistant dans le graphe D en examinant le moins d'arcs possible. Nous développons des bornes inférieures, des bornes supérieures et des règles de dominance pour réduire le nombre d'appels à AlgoBB. Nous essayons de choisir le bon arc dès le départ et prouvons ensuite l'optimalité de la solution.

L'approche proposée comporte les quatre points essentiels suivants :

1. Une fois que l'arc (i, j) a été traité [c'est-à-dire que le plus long chemin (D, G)-consistant passant par (i, j) a été calculé], l'arc (i, j) est supprimé du graphe D dans les itérations suivantes (lorsqu'on considère les autres arcs). En effet, aucune solution strictement meilleure P' passant par (i, j) ne peut exister et donc aucune solution passant par (i, j) ne peut être trouvée dans les itérations suivantes (le contraire contredirait l'optimalité de P).
2. Nous avons testé les règles suivantes pour l'ordre des arcs :
   (a) Ordre aléatoire.
   (b) D'abord les points d'articulation du graphe orienté D.
   (c) D'abord les arcs incidents aux sommets du centre du graphe orienté D.
   (d) D'abord les arcs qui ne sont pas présents dans les chemins (D, G)-consistants connus (le premier arc peut être choisi selon n'importe quelle règle, par exemple au hasard).
   (e) Les k premiers arcs aléatoirement, puis les arcs incidents aux sommets de la meilleure solution connue.
   (f) Les k premiers arcs aléatoirement, puis les points d'articulation du graphe D, et enfin tester les arcs incidents aux points d'articulation adjacents à la meilleure solution (D, G)-consistante connue.
   (g) D'abord les premiers arcs du plus long chemin dans D.
   (h) D'abord les arcs qui se trouvent entre les "communautés". Une communauté est un ensemble de sommets densément connectés. Comme les réseaux sociaux, les réseaux biologiques ont une structure communautaire naturelle (c'est-à-dire que les communautés peuvent être facilement identifiées). Pour identifier les communautés, nous utilisons l'algorithme de Girvan-Newman [Girvan and Newman, 2002], qui produit en séquence les arcs les plus probables entre les communautés. L'idée principale est de déconnecter rapidement de grandes parties de D pour maximiser l'efficacité de la partie $CoverSet$ de l'AlgoBB.
3. Le plus long chemin (D, G)-consistant passant par l'arc (i, j) est une borne inférieure pour les

itérations suivantes.

4. Le plus long chemin dans le graphe résiduel de D [c'est-à-dire le plus long chemin dans le graphe résultant de la suppression des arcs considérés dans les itérations précédentes ; cf. point 1], est une borne supérieure pour l'itération courante. En effet, dans le graphe résiduel, un chemin (D,G)-consistant passant par l'arc (i,j) est un chemin de ce graphe et ne peut être plus long que le plus long chemin de ce graphe.

Les résultats expérimentaux montrent que le meilleur algorithme en pratique combine les points 1, 2(h), 3, et 4. Nous présentons les résultat obtenues dans la section 3.7 (contraintes de chemin).

3.5 Modèles de programmation linéaire en nombres entiers

Dans cette section, nous présentons deux approches basées sur la programmation linéaire en nombres entiers pour résoudre le problème du One-To-One SkewGraM. Ces deux formulations comprennent deux ensembles de contraintes : Le premier ensemble correspond à la modélisation d'un chemin dans le graphe orienté $D = (V, A)$. Ce premier ensemble de contraintes est le même pour les deux modèles. Le second ensemble assure la connexité dans $G = (V, E)$. Nous proposons deux variantes de ce deuxième ensemble de contraintes, chacune basée sur une formulation différente du problème de l'arbre couvrant de poids minimal [Martin, 1991, Abdelmaguid, 2018, Captivo et al., 2009, Dias et al., 2017, Tilk and Irnich, 2018, Silvestri et al., 2017].

3.5.1 Chemin dans D : Contraintes de chemin communes aux deux modèles

Pour le graphe D, nous considérons dans les deux formulations deux sommets fictifs : un sommet source s et un sommet terminal t. Le sommet s est relié par des arcs sortants à tous les autres sommets dans D. Le sommet t est relié par des arcs entrants à tous les autres sommets dans D. Par souci de simplicité, nous définissons $A^+ = A \cup \{(s,i),\ \forall\ i \in V\} \cup \{(i,t),\ \forall\ i \in V\}$.

Les variables utilisées dans le premier ensemble de contraintes sont des variables binaires communes aux deux modèles et sont définies comme suit :

(a) $x_{ij} = 1$ si et seulement si l'arc $(i,j) \in A^+$ est présent dans le chemin ;
(b) $z_i = 1$ si et seulement si le sommet $i \in V$ est présent dans le chemin.

Comme nous recherchons le plus long chemin (D,G)-consistant, l'objectif est de maximiser le nombre d'arcs utilisés dans la solution (c'est-à-dire le nombre de variables x_{ij} fixées à 1),

$$\text{Max} \sum_{(i,j)\in A^+} x_{ij}.$$

La première contrainte limite la sortie de la source s à un seul arc,

$$\sum_{(s,i)\in A^+} x_{si} = 1. \tag{3.1}$$

La deuxième contrainte est la contrainte de conservation du flot,

$$\sum_{(i,k)\in A^+} x_{ik} = \sum_{(k,j)\in A^+} x_{kj},\ \forall\ k \in V. \tag{3.2}$$

La dernière contrainte définit la relation entre les variables z_i et x_{ij}. Le sommet $i \in V$ est dans le chemin ($z_i = 1$) si et seulement si un arc $(i,j) \in A^+$ part de ce sommet ($x_{ij} = 1$),

$$\forall\ i \in V,\ z_i = \sum_{(i,j)\in A^+} x_{ij}. \tag{3.3}$$

3.5.2 Connexité dans G : Contraintes de connexité

L'ensemble des sommets sélectionnés doit être connexe dans le graphe non-orienté G. Pour atteindre cet objectif, nous avons adapté deux formulations du problème de l'arbre couvrant de poids minimal [Martin, 1991, Abdelmaguid, 2018]. Pour rappel, pour un graphe non-orienté donné $\mathscr{W} = (\mathscr{N}, \mathscr{E})$, où $\mathscr{N}$ est l'ensemble des sommets, $\mathscr{E}$ est l'ensemble des arêtes, et l_{ij} est le poids de l'arête $(i,j) \in \mathscr{E}$, un arbre couvrant de poids minimal est un sous-ensemble d'arêtes $M \subseteq \mathscr{E}$ qui relie tous les sommets de $\mathscr{N}$ et tel que $\sum_{(i,j)\in M} l_{ij}$ est minimisé. Nous avons modifié ces formulations de sorte que les sommets sélectionnés dans D dans la section 4.3.1 (les sommets i tels que $z_i = 1$) forment un arbre.

3.5.2.1 Première variante (ILP 1)

Cette première variante est basée sur la formulation de Martin [Martin, 1991]. Nous définissons les variables booléennes suivantes :

(a) $y_{ij} = 1$ si l'arête $(i,j) \in E$ est prise dans l'arbre couvrant ;
(b) $b^k_{ij} = 1$ si l'arête $(i,j) \in E$ est dans l'arbre couvrant et le sommet $k \in V$ est du côté de j, ce qui signifie que k se retrouverait dans le sous-arbre de j si l'arête (i,j) était supprimée.

Les deux premières contraintes garantissent que, si l'arête $(i,j) \in E$ est sélectionnée, alors les sommets i et j sont sélectionnés,

$$z_i \geq y_{ij}, \ \forall\ (i,j) \in E, \tag{3.1}$$
$$z_j \geq y_{ij}, \ \forall\ (i,j) \in E. \tag{3.2}$$

La troisième contrainte impose que le nombre d'arêtes sélectionnées soit égal au nombre de sommets sélectionnés moins un,

$$\sum_{(i,j)\in E} y_{ij} = \sum_{i\in V} z_i - 1. \tag{3.3}$$

La quatrième contrainte garantit que, pour tous (i,j) dans E et k dans V, si (i,j) et k sont sélectionnés dans l'arbre ($y_{ij} = 1$, $z_i = 1$, $z_j = 1$, et $z_k = 1$), alors k doit être soit du côté de j ($b^k_{ij} = 1$) soit du côté de i ($b^k_{ji} = 1$). Si (i,j) n'est pas dans l'arbre ($y_{ij} = 0$), alors k n'est ni du côté de j ni du côté de i ($b^k_{ij} = b^k_{ji} = 0$),

$$b^k_{ij} + b^k_{ji} \leq y_{ij} + 1 - z_k, \ \forall\ (i,j) \in E, \ \forall\ k \in V. \tag{3.4}$$

Notez ici que, si un sommet k n'est pas sélectionné ($z_k = 0$), alors les variables b^k_{ij} sont sans importance dans le reste du modèle, et donc laissées libres.
Les deux dernières contraintes garantissent que, pour toute arête $(i,j) \in E$,

1. si l'arête (i,j) est sélectionnée dans l'arbre ($y_{ij} = 1$, $z_i = 1$, et $z_j = 1$), alors si une arête $(k,i) \in E$ connecte un sommet k à l'arbre, alors k ne peut pas être du côté de j (sinon, nous aurions un cycle contenant les sommets i, j, et k ; *cf.* Figure 3.2),
2. si les sommets i et j sont sélectionnés dans l'arbre, mais l'arête (i,j) n'est pas sélectionnée dans l'arbre ($y_{ij} = 0$, $z_i = 1$, et $z_j = 1$), il doit exister un sommet $k \in V$ tel que l'arête $(k,i) \in E$ et le sommet k sont dans l'arbre, et le sommet k est du côté de j (de sorte que i et j sont connectés dans l'arbre),

$$b^j_{ik} + y_{ij} \leq 2 - z_k, \quad \forall (i,j) \in E, \forall (k,i) \in E, \tag{3.5}$$

$$\sum_{(i,k)\in E} b^j_{ik} \geq z_i, \quad \forall\ (i,j) \notin E. \tag{3.6}$$

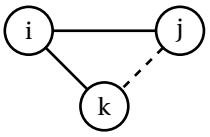


FIG. 3.2 : si l'arête (i, j) est sélectionnée dans l'arbre ($y_{ij} = 1$, $z_i = 1$, et $z_j = 1$), alors si une arête $(k, i) \in E$ connecte un sommet k à l'arbre, alors k ne peut pas être du côté de j (sinon, nous aurions un cycle contenant les sommets i, j, et k.

3.5.2.2 Deuxième variante (ILP 2)

Cette variante est basée sur une autre formulation du problème de l'arbre couvrant de poids minimal. Pour rappel, un arbre couvrant de poids minimal est équivalent à un ensemble plus courts chemins qui partent d'un sommet racine arbitraire et le connectent à tous les autres sommets du graphe [Abdelmaguid, 2018]. Cette formulation est basée sur un modèle qui trouve de tels chemins.
Un graphe orienté $(V, \overrightarrow{E})$ est construit en remplaçant chaque arête $a = (i, j) \in E$ dans $G = (V, E)$ par des arcs $(i, j), (j, i) \in \overrightarrow{E}$. Un sommet arbitraire est défini comme le sommet racine. Chaque sommet est associé à un niveau. La racine possède le niveau le plus bas qui est fixé à zéro tandis que les autres sommets doivent avoir des niveaux supérieurs à zéro. Le niveau d'un sommet est le nombre d'arêtes dans l'arbre entre la racine et ce sommet [Abdelmaguid, 2018]. (*cf.* Figure 3.3).

Étant donné que nous ne savons pas à l'avance quels seront les sommets de l'arbre, nous ne pouvons pas choisir arbitrairement une racine. Nous introduisons donc un sommet racine fictif ω qui est connecté par des arcs entrants et sortants à tous les autres sommets. Le niveau de ω est fixé à zéro. Par souci de simplicité, nous définissons $\overrightarrow{E^+} = \overrightarrow{E} \cup \{(\omega, i),\ \forall\ i \in V\} \cup \{(i, \omega),\ \forall\ i \in V\}$. Nous modifions cette formulation pour appliquer la condition selon laquelle les sommets sélectionnés dans la première partie du modèle (les sommets i tels que $z_i = 1$) forment un arbre. Nous utilisons le graphe orienté $\varphi = (V \cup \{\omega\}, \overrightarrow{E^+})$ et les variables de décision suivantes :

(a) variable binaire $y_{ij} = 1$ si et seulement si l'arc $(i, j) \in \overrightarrow{E^+}$ est pris dans l'arbre couvrant ;
(b) variable entière L_i, qui représente le niveau du sommet $i \in V \cup \{\omega\}$.

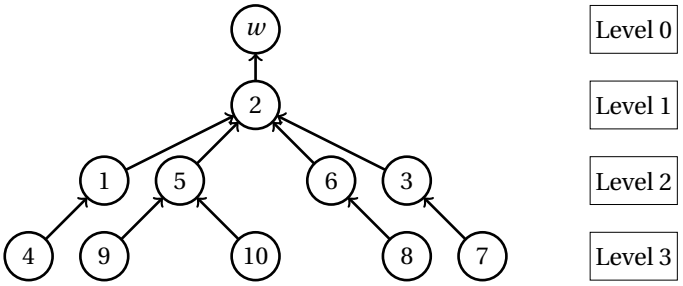


FIG. 3.3 : Exemple d'un arbre.

La première contrainte impose qu'un seul arc parte de chaque sommet sélectionné $i \in V$,

$$\sum_{(i,j) \in \overrightarrow{E^+}} y_{ij} = z_i,\ \forall\ i \in V. \tag{3.1}$$

La deuxième contrainte garantit qu'aucun arc ne part de la racine,

$$\sum_{(w,j) \in \overrightarrow{E^+}} y_{\omega j} = 0. \tag{3.2}$$

La troisième contrainte garantit que la racine est connectée à un arc,

$$\sum_{(i,\omega)\in\overrightarrow{E^+}} y_{i\omega} = 1. \tag{3.3}$$

La quatrième contrainte garantit que le niveau de la racine ω est égal à zéro,

$$L_w = 0. \tag{3.4}$$

Les cinquième et sixième contraintes définissent des bornes inférieures et supérieures pour les variables de décision de niveau,

$$L_i \leq \sum_{i\in V} z_i - 1,\ \forall\ i \in V, \tag{3.5}$$

$$L_i \geq 1,\ \forall\ i \in V. \tag{3.6}$$

La septième contrainte garantit que, pour une arête $(i,j) \in E$, seul un arc parmi (i,j) et $(j,i) \in \overrightarrow{E^+}$ peut être sélectionné,

$$y_{ij} + y_{ji} \leq z_i\ \forall\ (i,j) \in E. \tag{3.7}$$

La dernière contrainte définit la relation entre les variables de décision binaires y_{ij}, z_i et les variables de niveau correspondantes L_i et L_j. Elle impose la condition selon laquelle, si l'arc $(i,j) \in \overrightarrow{E^+}$ est sélectionné ($y_{ij} = 1$), le niveau du sommet i est supérieur d'une unité au niveau au sommet j ; sinon, cette contrainte est redondante,

$$L_i + 1 - z_i \geq L_j + y_{ij} - n(1 - y_{ij}),\ \forall\ (i,j) \in \overrightarrow{E^+}. \tag{3.8}$$

3.6 Modèle de programmation par contraintes

Dans les sections précédentes, nous avons présentés deux programmes linéaires efficaces pour résoudre le problème One-To-One SkewGraM. Dans cette section, nous présentons une formulation en contraintes (c'est-à-dire sous la forme d'un problème de satisfaction de contraintes, ou CSP) du problème One-To-One SkewGram, nous permettant de le résoudre à l'aide des outils de la Programmation par Contraintes (PPC).

A la frontière entre la recherche opérationnelle et de l'intelligence artificielle, la programmation par contraintes se démarque par son paradigme et sa polyvalence. L'approche est basée sur la réduction du domaine de décisions par la propagation des contraintes dans un arbre de branchement. L'efficacité de la programmation par contraintes se révèle pour des problèmes de décision dans lesquels il y a beaucoup d'interaction forte entre les différentes décisions. Chaque variable peut prendre un certain nombre de valeurs qui constituent son domaine. Les contraintes créent des relations entre les variables et limitent donc leurs domaines.

Un CSP est défini par un triplet (X, D, C) tel que

- X est un ensemble fini de variables ;
- D est une fonction qui fait correspondre à chaque variable $x_i \in X$ son domaine $D(x_i)$, c'est-à-dire l'ensemble fini des valeurs que peut prendre x_i ;
- C est un ensemble de contraintes, c'est-à-dire de relations entre certaines variables qui restreignent l'ensemble des valeurs qui peuvent être attribuées simultanément.

Une solution d'un CSP est une affectation de valeurs à toutes les variables de sorte que les domaines soient respectés et les contraintes soient satisfaites. Un CSP est dite inconsistant si l'ensemble de ses solutions est vide.

Le processus de résolution implique deux mécanismes : la propagation de contraintes, qui retire des valeurs incohérentes des domaines des variables à l'aide d'algorithmes de filtrage; l'exploration de l'espace de recherche, qui permet d'énumérer les combinaisons de valeurs admissibles lorsque le raisonnement apporté par la propagation n'est pas suffisant.

Les systèmes de programmation par contraintes sont conçus pour permettre la spécification et la résolution de modèles de contraintes. Les exemples typiques incluent des solveurs *open source* tels que Gecode [Chahira, 2013] et Choco [Prud'homme et al., 2017] et des solveurs commerciaux tels que SICStus Prolog [Carlsson, 2009] et CP Optimizer [Laborie et al., 2018].

Dans cette thèse nous évaluons nos modèles de programmation par contraintes en utilisant le solveur Choco [Prud'homme et al., 2017], qui est, à notre connaissance, le seul à proposer un module dédié à la résolution de problème de graphe.

Nous avons développé un modèle original de programmation par contraintes pour résoudre le problème One-To-One SkewGraM. Pour rappel, nous avons $D = (V, A)$ un graphe orienté, $G = (V, E)$ un graphe non-orienté et $n = |V|$ le nombre de sommets.

Ce modèle s'appuie sur des variables de type graphe [Fages, 2014]. Une variable de graphe g a un domaine défini par l'intervalle de graphes $D(g) = [\mathrm{g}, \overline{g}]$. Une instanciation g^* de g est un graphe satisfaisant la relation $\mathrm{g} \subseteq g^* \subseteq \overline{g}$. Le processus de résolution consiste alors à retirer de $\overline{g}$ certains noeuds et arêtes/arcs, ainsi qu'à ajouter à g des noeuds et arêtes/arcs. Ces étapes s'achèvent lorsque la variable graphe est instanciée, *i.e.*, lorsque $\mathrm{g} = \overline{g}$.

Une extension du solveur Choco pour les graphes [Bouamama et al., 2019] permet de déclarer des contraintes de graphe (le graphe doit être connexe, le graphe doit être un chemin, ...). Nous pressentons ci-dessous notre modèle nommé PPC1.

Les variables de décision sont définis comme suit :

- $d \in [\emptyset, D]$: une variable de type graphe orienté représentant le chemin de D.
- $g \in [\emptyset, G]$: une variable de type graphe non-orienté représentant le sous-graphe connexe de G contenant les sommets retenus.
- I et $J \in V$: deux variables de type entier représentant les sommets de début et de fin du chemin de D, respectivement.
- $z_i \in \{0, 1\}$: n variables booléennes indiquant si, $i \in V$ appartient au chemin retenu.
- $L \in \{2, \ldots, N\}$: une variable de type entier représentant la longueur du chemin retenu.

La maximisation de la longueur du chemin conduit à l'objectif suivant.

$$maximize(L)$$

Les contraintes sont formulées comme suit :

- La première contrainte garantit que d est un chemin de I à J de longueur L.

$$subPath(d, I, J, L) \tag{3.9}$$

- La second contrainte impose la connexité de g.

$$connected(g) \tag{3.10}$$

- Ces deux contraintes assurent que les sommets retenus dans d et g sont les mêmes.

$$nodesChanneling(g, \{z_i, i \in V\}) \tag{3.11}$$

$$nodesChanneling(d, \{z_i, i \in V\}) \tag{3.12}$$

Pour des raisons techniques liées à Choco et puisque la contrainte $subPath$ n'est plus maintenue pour les variables de type graphe, nous avons choisi de représenter le chemin de D par un ensemble de variable x_i, $i \in V$, représentant les successeurs de chaque sommet $i \in V$.

Avec ce nouveau choix, les variables de décision sont définis comme suit :

- $succ_i \in \{j \ inV, (i,j) \in A\} \cup \{i\} \cup \{n+1\}$: le successeur de $i \in V$ dans le chemin de D retenu.
  $x_i = j$ si j succède i dans le chemin retenu.
  $x_i = i$ si le sommet i n'est pas sélectionné dans le chemin retenu.
  $x_i = n+1$ si i est le dernier sommet du chemin retenu.
- $g \in g \in [\emptyset, G]$: une variable de type graphe non-orienté représentant le sous-graphe connexe de G contenant les sommets retenus.
- I et $J \in V$: deux variables de type entier représentant les sommets de début et de fin du chemin de D, respectivement.
- $z_i \in \{0,1\}$: n variables booléennes indiquant si, $i \in V$ appartient au chemin retenu.
- $L \in \{2,\ldots,N\}$: une variable de type entier représentant la longueur du chemin retenu.

La maximisation de la longueur du chemin conduit à l'objectif suivant.

$$maximize(L)$$

Les contraintes sont formulées comme suit :

- La première contrainte garantit que les variables $succ_i$ définissent bien un chemin de I à J de longueur L.
$$subPath(\{succ_i, i \in V\}, I, J, L) \tag{3.13}$$
- La second contrainte impose la connexité de g.
$$connected(g) \tag{3.14}$$
- Ces deux contraintes lient les variables z_i ($i \in V$), $succ_i$ ($i \in V$), et g. Elles imposent in fine que les sommets retenus dans $succ_i$ et g soient les mêmes.
$$nodesChanneling(g, \{z_i, i \in V\}) \tag{3.15}$$

$$succ_i = i \Longleftrightarrow z_i = 0, \forall i \in V \tag{3.16}$$

Ce modèle a été implémenté à l'aide du solveur Choco et testé sur des données aléatoires et des données réelles. Nous présenterons les résultats obtenus dans la section 3.7.

3.6.1 Borne supérieure

Dans la section ci-dessus, nous avons présenté un modèle de programmation par contraintes pour résoudre le problème One-To-One SkewGraM. Comme nous le verrons dans la section 3.7, les résultats que nous avons obtenus en testant ce modèle se sont avérés décevants. Le solveur ne disposant d'aucune borne sur la valeur de la fonction objectif, il peine en pratique à prouver l'optimalité de solutions qu'il trouve pourtant rapidement. Nous nous proposons maintenant de combler ce vide.

Pour cela, nous proposons de réduire le domaine de la variable L (la longueur du chemin dans D, c'est-à-dire notre fonction objectif) à l'aide d'une borne supérieure triviale calculée à partir des autres variables. En particulier, L sera toujours plus petit que le plus long chemin dans d. Autrement dit, on peut à tout instant majorer la valeur de L par la longueur du plus long chemin dans la borne supérieure du domaine de d. On calcule ainsi le plus long chemin dans D commençant dans le domaine de I à un sommet dans le domaine J. Comme D est acyclique, ce calcul peut être fait en temps polynomial.

Cette borne peut, en particulier, être calculée à l'aide de la procédure suivante :

– On calcule un ordre topologique des sommets de d, en ignorant les prédécesseurs de chaque potentiel premier sommet du chemin (des sommets dans le domaine courant de la variable I).
– Une fois que l'on a un tel ordre topologique, on peut appliquer l'algorithme de Bellman pour trouver les valeurs des plus longs chemins entre un potentiel premier sommet du chemin et tous les autres sommets.
– On en déduit la valeur d'un plus long chemin entre un potentiel premier sommet du chemin et un potentiel dernier sommet du chemin (un sommet apparaissant dans le domaine de la variable J).
– On supprime du domaine de L, toute valeur supérieure à la longueur d'un tel chemin.

Voire l'algorithme 1.

Cette procédure a été implémentée dans Choco au sein d'une contrainte :

$$longestPath(\{succ_i, i \in V\}, I, J, L) \tag{3.17}$$

Elle est donc appelée automatiquement, lorsque cela est nécessaire (lorsque les domaines de I, J et/ou d sont modifiés). Avec l'intégration de cette contrainte, notre modèle est nommé PPC2. Nous allons montrer les résultats de PPC2 dans la section 3.7.

Algorithm 1 Algorithme : La contrainte longestPath (redondante)

Données : Une variable $d = [\underline{d}, \overline{d}]$ de type graphe orienté et trois variables entières I, J et L.
Calculer un ordre topologique des sommets de $\overline{d}$ en ignorant les prédécesseurs des sommets apparaissant dans le domaine de I ;
Appliquer Bellman pour en déduire les valeurs des plus longs chemins de $\overline{d}$ entre un sommet apparaissant dans le domaine de I et tous les autres sommets ;
En déduire la valeur B d'un plus long chemin de $\overline{d}$ entre un sommet apparaissant dans le domaine de I et un sommet apparaissant dans le domaine de J ;
Mettre à jour la borne supérieure $L \leq B$;

3.7 Résultats expérimentaux

Pour montrer l'efficacité des méthodes proposées en termes de temps d'exécution et de qualité, nous les appliquons sur des graphes aléatoires de type Erdos–Renyi et scale-free.

Les réseaux d'Erdos–Renyi constituent le principal modèle de réseau pour l'étude des réseaux complexes du monde réel. Dans ces graphes, les noeuds ont approximativement le même nombre d'arêtes adjacentes.

Ces dernières années, les scientifiques ont découvert que la distribution des arêtes de nombreux réseaux du monde réel, tels que le World Wide Web, les réseaux sociaux et les réseaux métaboliques, suivent des distributions de type loi de puissance [Barabasi and Albert, 1999, Barabasi, 2009]. En 1999, Barabasi et Albert ont proposé des graphes scale-free avec des distributions d'arêtes de type loi de puissance [Barabasi and Albert, 1999].

Enfin, nous appliquons également les méthodes proposées à des réseaux biologiques, pour la recherche de chaînes de réactions dans un réseau métabolique catalysées par le produit de gènes voisins.

Pour les modèles linéaires, nous avons utilisé CPLEX 12.8.0 avec les paramètres par défaut et pour le modele de programmation par contraintes, nous avons utilisé Choco 4.10.5 [Prud'homme et al., 2017] avec les paramètres par défaut. Le temps d'exécution a été limité à 1300 s par instance pour les instances générées aléatoirement et à 300 s pour les données biologiques.

Dans tous les tableaux qui comparent les résultats pour des instances générées de façon aléatoire, les résultats de chaque méthode sont affichés dans deux colonnes : une colonne contient les temps d'exécution, et l'autre colonne contient les longueurs de solution. La longueur de la solution est en gras

si l'optimalité de la solution été prouvée. Dans tous les tableaux de résultats pour des données biologiques (voie métabolique versus génome), les résultats de chaque méthode sont présentées dans deux colonnes : une colonne pour le temps d'exécution, et une colonne pour le nombre d'instances où l'optimalité de le solution trouvée a été prouvée. Le temps est présenté en secondes.

3.7.1 Graphes d'Erdos–Renyi

Pour comparer les résultats des quatre méthodes (AlgoBB++, ILP1, ILP2 et PPC2), nous générons cinq instances pour chaque tuple (n, $p1$, $p2$), où :

(a) $n \in \{110, 460, 1060, 1500, 2000, 4000, 5000\}$ est le nombre de sommets ;
(b) $p1 \in \{0.05, 0.10, 0.20\}$ est la probabilité d'avoir un arc entre deux sommets pour D ;
(c) $p2 \in \{0.05, 0.10, 0.20\}$ est la probabilité d'avoir une arête entre deux sommets pour G.

Nous nous limitons dans ces expérimentations à présenter les instances les plus difficiles à résoudre à $p1 = 0.05$ et $p2 = 0.05$.

Le tableau 3.1 compare les performances des quatre méthodes proposées pour résoudre le problème. Dans le premier cas, $n = 110$, $p1 = 0.05$, et $p2 = 0.05$, et chaque méthode trouve la solution optimale. ILP2, AlgoBB++ et PPC2 sont similaires en termes de qualité et de temps d'exécution, alors que ILP1 nécessite un temps d'exécution plus long. Dans le troisième cas, $n = 1060$, $p1 = 0.05$, et $p2 = 0.05$, seuls AlgoBB++, ILP2 et PPC trouvent la solution optimale. Dans les deuxième, quatrième et cinquième cas, $n = 460$ ou $n = 1500$ ou $n = 2000$, $p1 = 0,05$, et $p2 = 0,05$, et PPC et ILP2 trouvent la solution optimale. La PPC est globalement beaucoup plus rapide. Dans le sixième cas, $n = 4000$, $p1 = 0,05$, et $p2 = 0,05$, seule PPC2 trouve la solution optimale. Dans le septième cas, $n = 5000$, $p1 = 0,05$, et $p2 = 0,05$, seule PPC2 trouve une solution où l'optimalité n'a pas été prouvée.

Nous remarquons que AlgoBB++ a passé le temps pour explorer les noeuds de l'arbre qui ne sont pas associés à des solutions alors que pour l'instance 1060 AlgoBB++ a tombé sur un noeud qui donné une solution et les règles de dominances AlgoBB++ trouve une solution optimale.

Dans cet ensemble de résultats numériques (graphes d'Erdos-Renyi), le nombre total d'instances est 35. En termes de qualité, ILP1 et AlgoBB++ trouvent respectivement 14 et 25 % des solutions optimales, alors que ILP2 et PPC2 trouvent respectivement 60 et 85 % des solutions optimales. Pour le temps d'exécution moyen, ILP1, AlgoBB++, ILP2 et PPC2 nécessitent 1135, 921, 657 et 347 s, respectivement. Notez qu'ILP1 échoue pour $n > 110$, AlgoBB++ échoue pour $n \geqslant 1500$, ILP2 échoue pour $n \geqslant 3000$ et PPC2 échoue pour $n \geqslant 5000$.

Ces résultats confirment que AlgoBB++ s'exécute plus rapidement et fournit une solution de meilleure qualité que ILP1. ILP2 s'exécute plus vite et fournit une une solution de meilleure qualité qu'AlgoBB++, et que PPC2 s'exécute plus vite et fournit une solution de meilleure qualité que ILP2.

Vous avez peut-être remarqué que nous ne rapportons pas les résultats obtenus en appelant AlgoBB m fois (i.e., une fois par arc–une approche très simple) et celles de PPC1. En fait, ces approches échouent mêmes pour les petites instances. Pour prouver cette affirmation, nous appliquons l'approche AlgoBB à cinq petites instances Erdos-Renyi où $n \in \{110, 160, 210\}$). Le tableau 3.2 compare les performances d'AlgoBB et d'AlgoBB++. AlgoBB atteint sa limite à $n = 210$.

Nous appliquons cette l'approche PPC1 à cinq petites instances Erdos-Renyi où $n \in \{110, 210, 360\}$). Le tableau 3.3 compare les performances de PPC1 et PPC2 (c'est-à-dire notre modèle de PPC, avec et sans la supérieure). PPC1 atteint sa limite à $n = 460$. Les résultats du tableau 3.3 confirme que PPC2 est la meilleure en temps et qualité par rapport à PPC1. A noter que dans les tableaux qui suivent, nous comparons PPC2 (c'est-à-dire modèle de PPC avec borne supérieure) avec les autres méthodes.

TAB. 3.1 : Instances de type Erdos–Renyi : $n \in \{110, 460, 1060, 1500, 2000, 4000, 5000\}$, $p1 = 0.05$, et $p2 = 0.05$. La longueur de solution en gras indique que la solution est optimale.

			AlgoBB++		ILP1			ILP2			PPC2	
Instance	$\|V\|$	$\|A\|$	$T(s)$	$\|Path\|$	$T(s)$	$\|Path\|$	Gap (%)	$T(s)$	$\|Path\|$	Gap (%)	$T(s)$	$\|Path\|$
1	110	294	11.08	**3**	141.78	**3**	0	3.99	**3**	0	0.31	**3**
2	110	298	1.99	**6**	192.17	**6**	0	3.24	**6**	0	0.22	**6**
3	110	291	1.26	**3**	183.90	**3**	0	3.77	**3**	0	0.27	**3**
4	110	295	0.62	**3**	117.16	**3**	0	3.80	**3**	0	0.23	**3**
5	110	305	0.46	**4**	112.68	**4**	0	3.51	**4**	0	0.23	**4**
6	460	5271	1300	—	1300	—	—	1287.66	**40**	0	26.39	**40**
7	460	5275	1300	—	1300	—	—	707.21	**40**	0	61.97	**40**
8	460	5226	1300	—	1300	—	—	695.05	**38**	0	15.31	**38**
9	460	5209	1300	—	1300	—	—	1300	40	2.44	29.33	**40**
10	460	5273	1300	—	1300	—	—	1300	43	4.55	46.94	**43**
11	1060	28105	924.49	**102**	1300	—	—	148.63	**102**	0	10.15	**102**
12	1060	27993	911.34	**94**	1300	—	—	277.62	**94**	0	18.75	**94**
13	1060	27734	1300	102	1300	—	—	88.47	**103**	0	11.15	**103**
14	1060	28114	860.84	**99**	1300	—	—	145.87	**99**	0	22.53	**99**
15	1060	27981	949.61	**90**	1300	—	—	95.06	**90**	0	367.14	**90**
16	1500	55996	1300	—	1300	—	—	133.18	**135**	0	21.91	**135**
17	1500	56173	1300	—	1300	—	—	377.44	**140**	0	15.78	**140**
18	1500	56275	1300	—	1300	—	—	122.98	**140**	0	26.55	**140**
19	1500	56160	1300	—	1300	—	—	126.87	**137**	0	24.92	**137**
20	1500	56374	1300	—	1300	—	—	113.58	**145**	0	43.31	**145**
21	2000	99946	1300	—	1300	—	—	1300	3	—	73.50	**178**
22	2000	100201	1300	—	1300	—	—	1105.17	**82**	0	131.25	**182**
23	2000	99679	1300	—	1300	—	—	352.19	**195**	0	126.74	**195**
24	2000	100187	1300	—	1300	—	—	334.58	**179**	0	92.68	**179**
25	2000	100395	1300	—	1300	—	—	1300	2	—	59.10	**182**
26	4000	400219	1300	—	1300	—	—	1300	—	—	1148.50	**379**
27	4000	400356	1300	—	1300	—	—	1300	—	—	739.57	**379**
28	4000	400022	1300	—	1300	—	—	1300	—	—	866.77	**374**
29	4000	400793	1300	—	1300	—	—	1300	—	—	656.37	**384**
30	4000	400130	1300	—	1300	—	—	1300	—	—	1024.69	**378**
31	5000	626123	1300	—	1300	—	—	1300	—	—	1300	413
32	5000	624474	1300	—	1300	—	—	1300	—	—	1300	430
33	5000	625315	1300	—	1300	—	—	1300	—	—	1300	402
34	5000	624471	1300	—	1300	—	—	1300	—	—	1300	420
35	5000	624898	1300	—	1300	—	—	1300	—	—	1300	413

TAB. 3.2 : Instances de type Erdos–Renyi : $n \in \{110, 160, 210\}$, $p1 = 0.05$, et $p2 = 0.05$. La longueur de solution en gras indique que la solution est optimale.

			AlgoBB		AlgoBB++	
Instance	$\|V\|$	$\|A\|$	$T(s)$	$\|Path\|$	$T(s)$	$\|Path\|$
1	110	298	17.20	**6**	1.99	**6**
2	110	291	9.16	**3**	1.26	**3**
3	160	651	930.95	**7**	118.97	**7**
4	160	685	687.13	**7**	203.48	**7**
5	210	1119	1300	5	853.50	**7**

TAB. 3.3 : Instances de type Erdos–Renyi : $n \in \{110, 210, 360, 460\}$, $p1 = 0.05$, et $p2 = 0.05$. La longueur de solution en gras indique que la solution est optimale.

			PPC1		PPC2	
Instance	$\|V\|$	$\|A\|$	$T(s)$	$\|Path\|$	$T(s)$	$\|Path\|$
1	110	297	0.206	**3**	0.198	**3**
2	110	301	0.192	**6**	0.169	**6**
3	110	293	0.190	**3**	0.175	**3**
4	110	298	0.172	**3**	0.192	**3**
5	110	308	0.156	**4**	0.199	**4**
6	210	1081	1.523	**11**	1.518	**11**
7	210	1121	1.145	**7**	1.183	**7**
8	210	1136	1.680	**9**	2.030	**9**
9	210	1101	1.834	**7**	2.044	**7**
10	210	1083	0.992	**8**	1.090	**8**
11	360	3193	8.287	**30**	8.287	**30**
12	360	3182	721.984	**30**	20.015	**30**
13	360	3250	02.180	**29**	23.575	**29**
14	360	3155	340.323	**29**	25.804	**29**
15	360	3247	587.053	**29**	47.765	**29**
16	460	5272	1300	40	39.75	**40**
17	460	5276	1300	40	90.61	**40**
18	460	5227	1300	38	21.58	**38**
19	460	5210	1300	40	47.99	**40**
20	460	5217	1300	43	86.96	**43**

3.7.2 Graphes de scale-free

Étant donné que des études récentes montrent qu'un réseau biologique tend à être scale-free [Barabasi, 2009], nous appliquons également les méthodes proposées sur des graphes scale-free générés aléatoirement. Dans cette expérimentation, nous avons généré cinq instances pour chaque valeur de n, où $n \in \{110, 460, 1060, 4000, 6000, 40000\}$ est le nombre de sommets.

Le tableau 3.4 compare les résultats des quatre méthodes proposées pour résoudre le problème. Dans le premier cas, $n = 110$ et chaque méthode trouve la solution optimale. ILP2, AlgoBB++ et PPC2 sont similaires en termes de qualité de solution et de temps d'exécution, alors que ILP1 nécessite un temps d'exécution plus long. Dans le deuxième cas, $n = 460$ et ILP1, ILP2 et PPC2 trouvent la solution optimale. Dans les troisième ($n = 1060$) et quatrième ($n = 4000$) cas, seuls ILP2 et PPC2 trouvent la solution optimale. La PPC2 est globalement beaucoup plus rapide. Dans le cinquième cas, $n = 6000$, $p1 = 0,05$, et $p2 = 0,05$, seule PPC2 trouve la solution optimale. Dans le sixième cas, $n = 40000$, $p1 = 0,05$, et $p2 = 0,05$, seule PPC2 trouve une solution sans garantie qu'elle est optimale.

Dans cet ensemble de résultats numériques (graphes scale-free), le nombre total d'instances est 30. En termes de qualité, ILP1 et AlgoBB++ trouvent respectivement 16 et 45 % des solutions optimales, alors que ILP2 et PPC2 trouvent respectivement 63 et 83 % des solutions optimales. Pour le temps d'exécution moyen, ILP1, AlgoBB++, ILP2 et PPC2 nécessitent 1095, 766, 526 et 220 s, respectivement. Notez qu'ILP1 échoue pour $n > 110$, AlgoBB++ échoue pour $n \geqslant 4000$, ILP2 échoue pour $n \geqslant 6000$ et PPC2 échoue pour $n \geqslant 40000$.

Ces résultats confirment que AlgoBB++ s'exécute plus rapidement et fournit une solution de meilleure qualité que ILP1., ILP2 s'exécute plus vite et fournit une une solution de meilleure qualité qu'AlgoBB++, et que PPC2 s'exécute plus vite et fournit une solution de meilleure qualité que ILP2.

3.7.3 Données biologiques

Un réseau métabolique est représenté dans la base de données KEGG comme une collection de modules fonctionnels (c'est-à-dire de petits réseaux) appelés voies métaboliques. Par conséquent, les voies métaboliques peuvent être modélisées par des graphes acycliques orientés [Wernicke and Rasche, 2007].

Le génome est modélisé par un graphe non-orienté dont les sommets représentent les gènes et une arête sépare deux gènes adjacents. Nous avons récupéré les informations métaboliques et génomiques de 50 espèces dans la base de données KEGG (l'encyclopédie des gènes et des génomes de Kyoto). Nous avons considéré toutes les voies de chaque espèce. Cela donne un total de 5378 instances. Le temps limite pour chaque instance est de 300 secondes.

Comme détaillé dans [Babou, 2012], le graphe D est la voie métabolique que nous voulons étudier, et le graphe G est un graphe non-orienté construit à partir du réseau du génome sur le même ensemble de sommets que D. Nous appliquons les méthodes proposées pour extraire automatiquement d'une voie métabolique une chaîne de réactions qui sont catalysées par les produits de gènes voisins dans le génome (chemin dans D dont l'ensemble des sommets induit un sous-graphe connexe dans G).

Comme dans une approche de la littérature basée sur les graphes pour l'intégration de données biologiques hétérogènes dans un autre contexte [Boyer et al., 2005], une étape de prétraitement a été ajoutée afin de tenir compte des réactions et/ou des gènes non contigus. L'étape de prétraitement consiste à modifier les graphes d'entrée en ajoutant des arcs (respectivement des arêtes) entre les sommets séparés par au plus δ_D autres réactions (respectivement δ_G autres gènes) et par conséquent D et/ou G sont plus denses. Les valeurs de $\delta_D \in \{0, 1, 2, 3\}$ et/ou $\delta_G \in \{0, 1, 2, 3\}$ doivent être fixées assez bas (par exemple, au plus 3) pour garantir que les *trails* produits sont pertinents d'un point de vue biologique.

Par exemple, les arêtes pleines noires correspondant à $\delta_G = 0$ relient les gènes A à E dans le graphe non-orienté G de la Figure 3.4. Un gène peut être omis si δ_G est fixé à 1, dans ce cas l'ensemble des arêtes de G comprend les arêtes noires en pointillés. Enfin, si deux gènes peuvent être omis ($\delta_G = 2$),

TAB. 3.4 : Instances de scale-free : $n \in \{110, 460, 1060, 4000, 6000, 10000, 40000\}$. La longueur de solution en gras indique que la solution est optimale.

			AlgoBB++		ILP1			ILP2			PPC2	
Instance	$\|V\|$	$\|A\|$	$T(s)$	$\|Path\|$	$T(s)$	$\|Path\|$	Gap (%)	$T(s)$	$\|Path\|$	Gap (%)	$T(s)$	$\|Path\|$
1	110	179	2.94	**7**	67.12	**7**	0	4.84	**7**	0	0.03	**7**
2	110	179	0.67	**10**	66.11	**10**	0	1.23	**10**	0	0.04	**10**
3	110	155	0.49	**9**	61.80	**9**	0	1.67	**9**	0	0.03	**9**
4	110	190	2.84	**11**	58.84	**11**	0	4.84	**11**	0	0.04	**11**
5	110	167	1.12	**8**	105.50	**8**	0	2.42	**8**	0	0.05	**8**
6	460	796	93.58	**14**	1300	—	—	4.84	**14**	0	0.23	**14**
7	460	735	11.49	**17**	1300	—	—	9.69	**17**	0	0.15	**17**
8	460	748	28.32	**14**	1300	—	—	4.84	**14**	0	0.10	**14**
9	460	759	5.87	**14**	1300	—	—	1.45	**14**	0	0.17	**14**
10	460	714	92.78	**13**	1300	—	—	8.82	**13**	0	0.17	**13**
11	1060	1732	6.40	**20**	1300	—	—	16.84	**20**	0	0.35	**20**
12	1060	1732	119.82	**18**	1300	—	—	16.49	**18**	0	0.35	**18**
13	1060	1796	538.24	**20**	1300	—	—	49.27	**20**	0	0.62	**20**
14	1060	1699	1300	16	1300	—	—	9.69	**18**	0	0.46	**18**
15	1060	1700	1300	18	1300	—	—	4.87	**19**	0	0.60	**19**
16	4000	7348	1300	24	1300	—	—	266.96	**32**	0	2.24	**32**
17	4000	7051	1300	17	1300	—	—	1300	26	3.70	3.41	**26**
18	4000	7572	1300	28	1300	—	—	1300	29	3.30	2.11	**29**
19	4000	7150	1300	19	1300	—	—	249.33	**28**	0	1.80	**28**
20	4000	7080	1300	25	1300	—	—	192.86	**27**	0	2.11	**27**
21	6000	10997	1300	25	1300	—	—	1300	2	—	9.31	**34**
22	6000	10676	1300	25	1300	—	—	1300	36	2.11	7.45	**36**
23	6000	11118	1300	25	1300	—	—	1300	33	2.78	16.65	**33**
24	6000	11287	1300	25	1300	—	—	1300	35	2.98	6.76	**35**
25	6000	11357	1300	25	1300	—	—	658.86	**31**	0	51.39	**31**
26	40000	75176	1300	25	1300	—	—	1300	—	—	1300	48
27	40000	76363	1300	25	1300	—	—	1300	—	—	1300	55
28	40000	76437	1300	25	1300	—	—	1300	—	—	1300	59
29	40000	75165	1300	25	1300	—	—	1300	—	—	1300	60
30	40000	76881	1300	25	1300	—	—	1300	—	—	1300	55

l'ensemble des arêtes de G comprend également les arêtes bleues en pointillés.

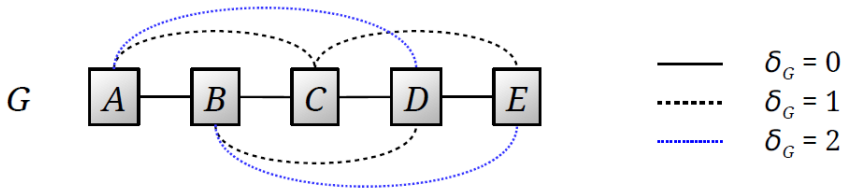


FIG. 3.4 : Illustration du paramètre gap. Si δ_G est positif, des arêtes supplémentaires doivent être ajoutées à G.

Les tableau 3.5 présente les résultats des quatre méthodes proposées pour résoudre le problème. En ce qui concerne la qualité des résultats, AlgoBB++, ILP1 et PPC trouvent 100% des solutions optimales, ILP2 trouve 99.8% des solutions optimales. En termes de temps d'exécution, ILP1 est plus rapide que ILP2, AlgoBB++ est plus rapide que ILP1, et PPC est plus rapide que AlgoBB++.

TAB. 3.5 : Instances biologiques : Voie métabolique versus génome

5378 instances de 50 espèces

	AlgoBB++		ILP1		ILP2		PPC2	
Number of instances	$T(s)$	Optimal	$T(s)$	Optimal	$T(s)$	Optimal	$T(s)$	Optimal
5378	4036	5378	7876	5378	16708	5370	6391	5378

3.7.3.1 Exemple de chemin

Nous présentons dans cette section des figures illustrant le regroupement des chemins (une chaines de réactions catalysées par le produits des gènes voisins).

Le système de nomenclature des gènes d'E. coli utilise les identificateurs de Blattner ou $b\ numbers$. Les $b\ numbers$ consécutifs reflètent généralement des gènes voisins (par exemple, les gènes b0086 et b0087).

Le système de dénomination des gènes de B. subtilis a la forme $BSUXXXX0$, où X est un chiffre. Les incréments de 10 dans les identifiants des gènes de B. subtilis reflètent généralement des gènes voisins (par exemple, les gènes BSU28300 et BSU28310 sont consécutifs).

La figure 3.5 montre une chaine de réactions de l'espèce E. *coli* dans la voie de $Pentose\ and\ glucuronate\ interconversions$. La chaine de réactions R07127 → R07125 → R03244 → R05850 est catalysée par le produits des gènes $\{eco:b3580, eco:b3581, eco:b3582, eco:b3583\}$.

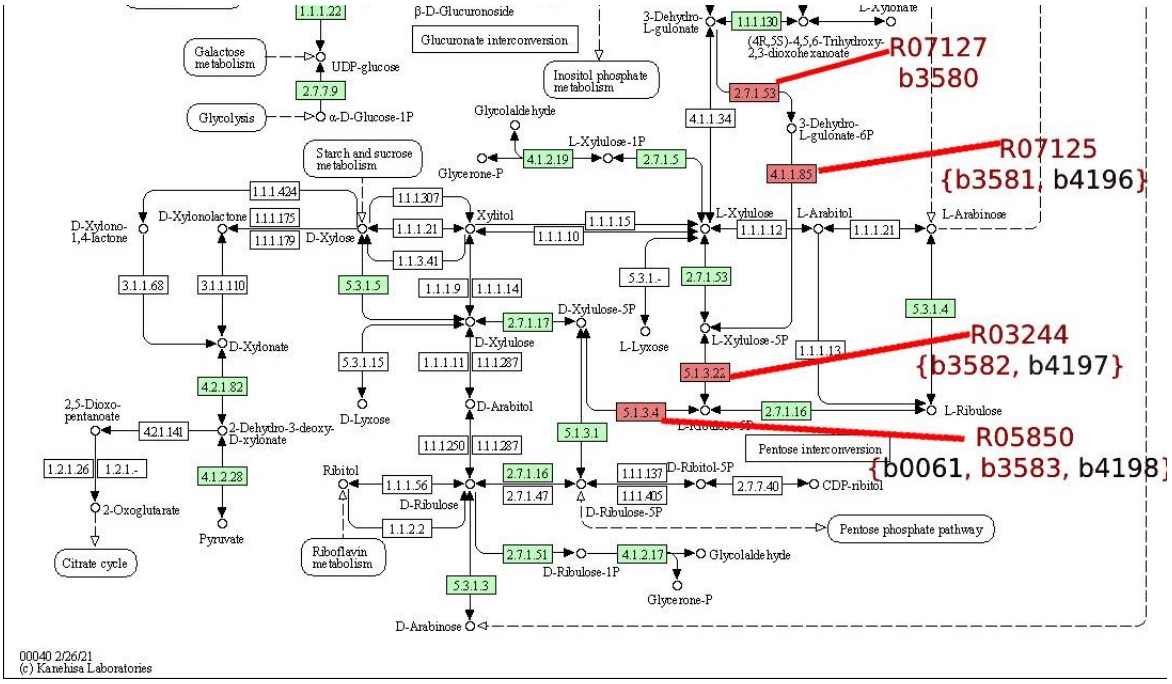


FIG. 3.5 : Une chaine de réactions catalysée par le produits des gènes voisins de l'espèce E. coli dans la voie métabolique *Pentose and glucuronate interconversions*. La chaine $R07127 \rightarrow R07125 \rightarrow R03244 \rightarrow R05850$ sont catalysées par le produits des gènes $\{eco : b3580, eco : b3581, eco : b3582, eco : b3583\}$. On voit ici une chemin constitué par les réactions avec une couleur de fond rouge. Les réactions chemin sont étiquetées avec les identifiants de réaction KEGG correspondants ($R\ numbers$) et avec les identifiants des gènes impliqués dans les réactions. Les gènes avec des identificateurs rouges sont voisins.

La figure 3.6 montre une chaine de réactions de l'espèce *Y. pestis* dans la voie de $Pentose\ and\ glucuronate\ interconversions$. Pour produire la chaine de réactions $R02439 \rightarrow R01761 \rightarrow R01526$ catalysées par les produits des gènes $\{ype : YPO2254, ype : YPO2253, ype : YPO2254\}$, la réaction $R01529$ réalisant l'activité enzymatique $EC : 5.1.3.1$ a été omise ($\delta_D = 1$).

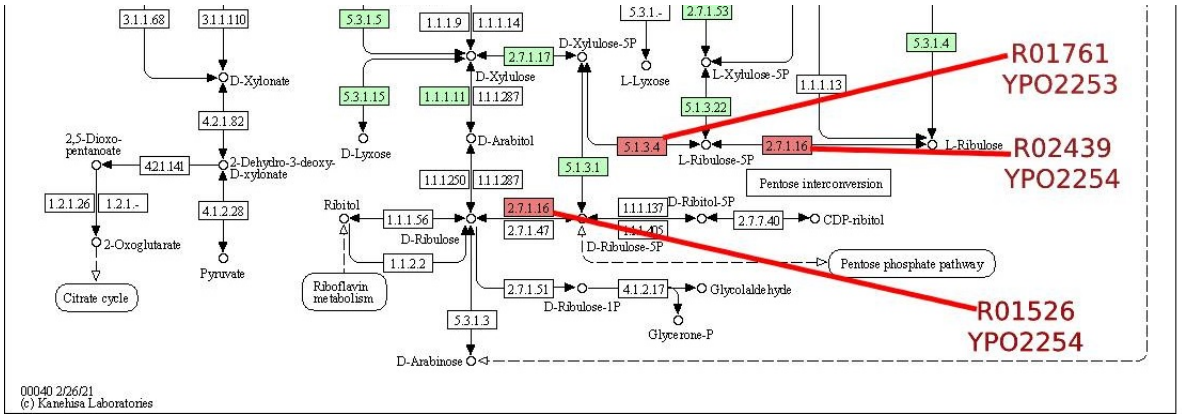


FIG. 3.6 : La chaine de réactions $R02439 \rightarrow R01761 \rightarrow R01526$ est catalysées par les produits des gènes $\{ype : YPO2254, ype : YPO2253, ype : YPO2254\}$. On voit ici une chemin constitué par les réactions avec une couleur de fond rouge. Les réactions chemin sont étiquetées avec les identifiants de réaction KEGG correspondants (*R numbers*) et avec les identifiants des gènes impliqués dans les réactions. Les gènes avec des identificateurs rouges sont voisins.

La figure 3.7 montre une chaine de réactions de l'espèce *Y. pestis* dans la voie de *Pentose and glucuronate interconversions*. La chaine de réactions $R04617 \rightarrow R03193 \rightarrow R02783$ est catalysées par les produits des gènes $\{vco : VC0395\_A1983, vco : VC0395\_A1978, vco : VC0395\_A1981\}$. les gènes $\{vco : VC0395\_A1979, vco : VC0395\_A1980, vco : VC0395\_A1982\}$ ont été omis ($\delta_G = 3$).

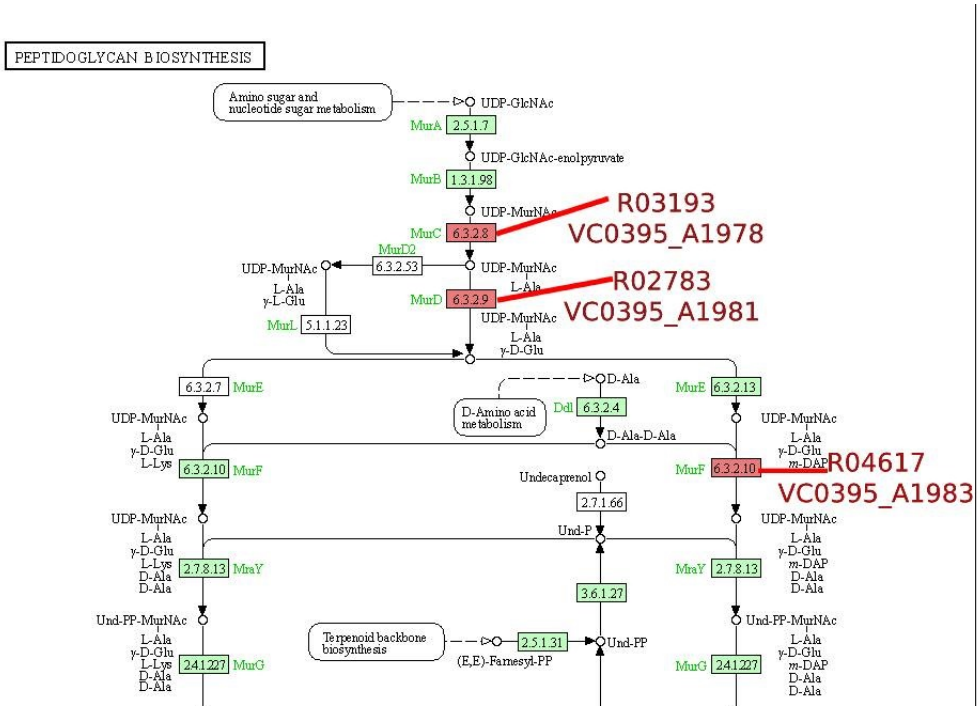


FIG. 3.7 : Une chaine de réactions catalysée par le produits des gènes voisins de l'espèce *Y. pestis* dans la voie métabolique *Pentose and glucuronate interconversions*. La chaine de réactions $R04617 \rightarrow R03193 \rightarrow R02783$ est catalysée par les produits des gènes $\{vco : VC0395\_A1983, vco : VC0395\_A1978, vco : VC0395\_A1981\}$. Les gènes $vco : VC0395\_A1979, vco : VC0395\_A1980, vco : VC0395\_A1982$ ont été omis ($\delta_G = 3$). On voit ici une chemin constitué par les réactions avec une couleur de fond rouge. Les réactions chemin sont étiquetées avec les identifiants de réaction KEGG correspondants ($R\ numbers$) et avec les identifiants des gènes impliqués dans les réactions. Les gènes avec des identificateurs rouges sont voisins.

3.8 Conservation du chemin

Dans les sections précédentes, des méthodes ont été présentées pour trouver les chemins des réactions qui sont catalysées par les produits de gènes voisins pour une espèce donnée. La présente section illustre comment ces chemins peuvent être exploités pour étudier la conservation des voies métaboliques et génomiques entre différentes espèces. La voie métabolique est représentée par un graphe biparti dans lequel les sommets peuvent être séparés en 2 groupes (métabolites et réactions).

Nous nous concentrons sur deux espèces, à savoir *Escherichia coli* et *Yersinia pestis*, et montrons que les chemins extraits dans l'espèce *Escherichia coli* peuvent être totalement conservés (trait commun), partiellement conservés (peut-être la résultat d'une évolution) ou non conservés (différence) dans l'espèce *Yersinia pestis*. Un chemin est conservé entre deux espèces s'il existe et s'il est constant dans les deux espèces. Un chemin est partiellement conservé si seulement un sous-chemin DG-consistant de celui-ci, existe dans la seconde espèce. Nous analysons les résultats obtenus par AlgoBB++, qui s'est avéré être la méthode la plus efficace pour de telles instances.

Dans la voie *Métabolisme des acides dibasiques ramifiés en C5* de l'espèce *Escherichia coli*, AlgoBB++ a trouvé le chemin des réactions $R00994 \rightarrow R03898 \rightarrow R03896$, catalysées par les produits des gènes voisins $\{eco : b0073, eco : b0072, eco : b0071\}$. Dans l'espèce *Yersinia pestis*, AlgoBB++ a trouvé le même chemin de réactions $R00994 \rightarrow R03898 \rightarrow R03896$, catalysé par les produits des gènes voisins $\{ype : YPO0532, ype : YPO0531, ype : YPO0530\}$. Nous pouvons alors conclure que ce chemin est totalement

conservé entre les deux espèces (*cf.* Figure 3.8).

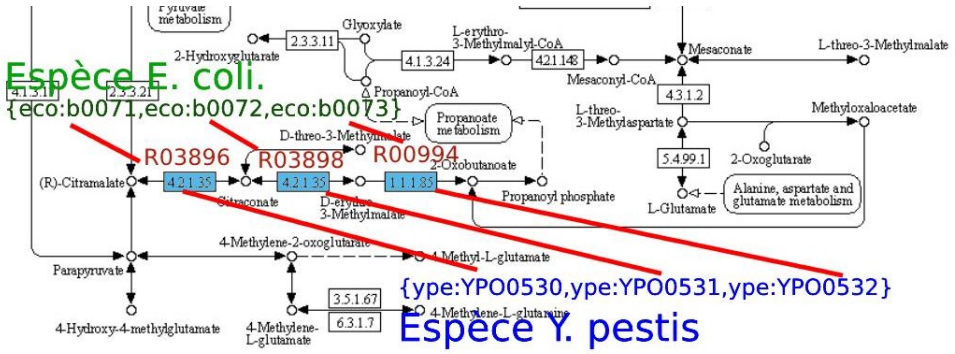


FIG. 3.8 : Chemin totalement conservé (trait commun)

Dans la voie *Métabolisme du galactose* de l'espèce *Escherichia coli*, AlgoBB++ a trouvé le chemin des réactions $R01092 \rightarrow R00955 \rightarrow R00291$, catalysées par les produits des gènes voisins $\{eco : b0757, eco : b0758, eco : b0759\}$. Dans l'espèce *Yersinia pestis*, AlgoBB++ a trouvé un sous chemin de réactions $R00955 \rightarrow R00291$, catalysé par les produits des gènes voisins $\{ype : YPO1138, ype : YPO1139\}$. Nous pouvons alors conclure que ce chemin est partiellement conservé entre les deux espèces (*cf.* Figure 3.9).

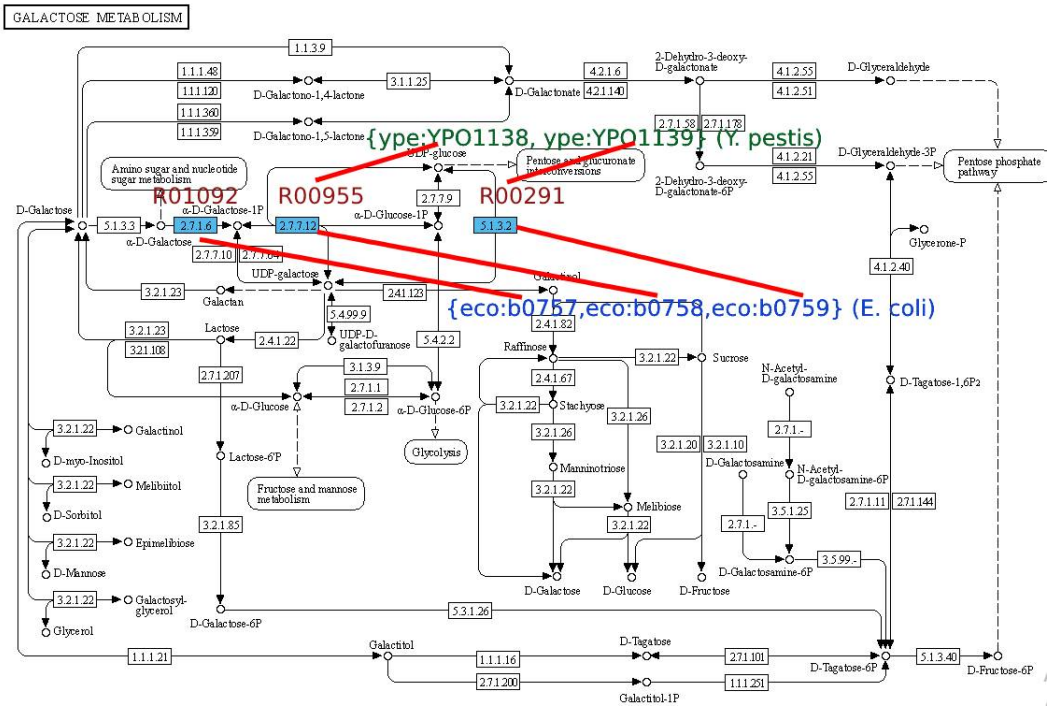


FIG. 3.9 : Chemin partiellement conservé (peut-être la résultat d'une évolution)

Dans la voie *Pentose phosphate pathway* de l'espèce Escherichia coli, AlgoBB++ a trouvé le chemin des réactions $R01827 \rightarrow R01830 \rightarrow R01641$, catalysées par les produits des gènes voisins $\{eco : b2464, eco : b2465, eco : b2465\}$. Chez l'espèce *Yersinia pestis*, il n'existe pas de solution pour cette voie.

Nous pouvons donc conclure que cette voie n'est pas conservée entre les deux espèces.

Une telle étude peut permettre d'extraire de nouvelles connaissances et/ou de confirmer et/ou d'infirmer les connaissances issues de la phylogénie. Une étude comparative plus approfondie nécessite une expertise biologique sera présentée dans le chapitre 5.

3.9 Conclusion

Dans ce chapitre, nous avons présenté une étude d'un problème d'optimisation combinatoire appelé "One-To-One SkewGraM". La complexité de ce problème est $\mathcal{NP}$-difficile au sens fort. Ce problème a des applications dans l'analyse des réseaux biologiques et des applications prévisibles dans l'analyse des réseaux sociaux, d'information et de communication.

Pour résoudre ce problème, nous avons proposé une généralisation d'une méthode de branch-and-bound appelée AlgoBB++, deux modèles linéaires, un modèle de programmation par contraintes et avons comparé les résultats de ces méthodes en termes de temps d'exécution et de qualité en les appliquant sur des graphes aléatoires et des données biologiques.

Les résultats numériques pour les instances générées aléatoirement ont révélé l'efficacité des méthodes. Ces résultats ont confirmé que, pour les grandes instances, AlgoBB++ s'exécute plus rapidement et trouve plus de résultats que ILP1, ILP2 s'exécute plus rapidement et trouve plus de solutions que AlgoBB++, et que PPC2 s'exécute plus rapidement et trouve plus de résultats que ILP2.

Les résultats numériques pour les données biologiques ont révélé l'efficacité des méthodes proposées. Pour ces petites instances, PPC, AlgoBB++ et ILP1 ont trouvé 100% des solutions optimales, et ILP2 a trouvé 99,8% des solutions optimales. Le temps d'exécution total pour ILP2, ILP1, PPC et AlgoBB++ était de 16708, 7876, 6391 et 4036 s, respectivement. Ces résultats confirment que, pour les petites instances, ILP1 s'exécute plus rapidement que ILP2, PPC s'exécute plus rapidement que ILP1, et que AlgoBB++ s'exécute plus rapidement que PPC.

Les résultats présentés améliorent nettement les meilleurs résultats de la littérature. Ces travaux ont été publié dans un article [Ahmed Sidi et al., 2022].

Chapitre 4

Le problème SkewGraM

Contents

4.1 Introduction

Dans le chapitre précédent, nous avons supposé que le graphe D en entrée du problème One-to-One SkewGraM est acyclique. Cette hypothèse est due à des motivations algorithmiques et biologiques (les *voies* métaboliques peuvent être modélisées par des DAGs). En revanche, en général les *réseaux* métaboliques contiennent des cycles. Donc, pour que les approches proposées soient applicables sur ces *réseaux* (et non pas seulement sur des *voies* métaboliques), la présence des cycles doit être prise en considération.

Dans ce chapitre, nous étudions le cas où le graphe modélisant le réseau métabolique peut contenir des circuits. Nous allons proposer des méthodes exactes pour la résolution d'un cas plus général du problème, qui consiste à calculer le *trail* maximal de réactions dans un réseau métabolique, de sorte que ces réactions soient toutes catalysées par des produits de gènes voisins. Autrement dit, nous cherchons un *trail* de couverture maximale T dans un graphe orienté $D = (V, A)$, tel que les sommets du *trail* T induisent un sous-graphe connexe d'un graphe non-orienté $G = (V, E)$ (les deux graphes sont construits sur le même ensemble de sommets). Ces *trails* sont utilisés par les biologistes pour comparer les espèces.

Comme dans le chapitre précédent, nous commencerons par formaliser ce problème, puis nous proposerons des méthodes de résolution adoptées, et des expérimentations sur des données aléatoires et des données réelles.

4.2 Définition du problème

Zaharia *et al.* [Zaharia et al., 2019] ont traité le problème présentée dans la section précédente où le graphe D est cyclique. La grande majorité des voies métaboliques présentent des cycles (*e.g.*, des réactions réversibles). L'objectif est de chercher une chaîne de réactions qui sont catalysées par les produits de gènes voisins. Une ou plusieurs réactions peuvent donc apparaitre plus d'une fois dans une solution. Pour capturer les cycles dans les solutions, nous cherchons un chemin non-élémentaire, dit *trail*. Nous définissons maintenant le concept de couverture d'un *trail* qui est le nombre de sommets distincts présents dans un *trail*. La nouvelle formulation du problème fournit des *trails* comme solutions, au lieu de chemins. Ce problème est formulé comme suit.

Soit $D = (V, A)$ un graphe orienté quelconque et $G = (V, E)$ un graphe non-orienté construit sur le même ensemble de sommets $V = \{1, 2, \ldots, n\}$. Un *trail* (D, G)-consistant T est un chemin (dirigé), élémentaire ou non, dans D tel que les sommets de T induisent un sous-graphe connexe dans G. Le problème consiste à calculer le *trail* (D, G)-consistant de couverture maximale (*cf.* Figure 4.1). Le cas particulier où D est acyclique (One-to-One SkewGraM) étant déjà $\mathcal{NP}$-difficile au sens fort, le problème général l'est également. Du point de vue algorithmique et pour résoudre le cas général, Zaharia *et al.* ont développé une méthode exacte appelée *HNet*. La méthode *HNet* prend en arguments un graphe orienté qui peut contenir des cycles $D = (V, A)$, un graphe non-orienté $G = (V, E)$, et un arc $(i, j) \in A$. Cet algorithme calcule le *trail* (D, G)-consistant de couverture maximale passant par un arc (i, j). Pour trouver la solution optimale, $HNet$ est appelé $|A| = m$ fois (un nouvel arc est considéré à chaque itération). HNet énumère des *trails* dans une voie métabolique (graphe orienté D), reposant sur l'énumération des chemins dans le *line graph* associé (graphe orienté $L(D)$). Le *line graph* orienté $L(D)$ représente le *line graph* du graphe orienté D. Par définition du *line graph*, les sommets de $L(D)$ sont des arcs de D.

Dans la suite du chapitre, nous proposons deux formulations de programmation linéaire en nombres entiers (ILP) et une méthode de programmation par contraintes pour résoudre ce nouveau problème. Ces trois formulations sont des adaptations de celles présentées dans le chapitre 3.

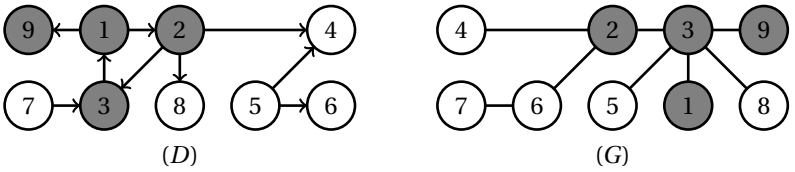


FIG. 4.1 : Exemple. Les *trail* $1 \to 2 \to 3 \to 1 \to 9$ et $3 \to 1 \to 2 \to 4$ sont (D, G)-consistants et couvrent quatre sommets distincts. En revanche, le *trail* de couverture maximale dans D est $7 \to 3 \to 1 \to 2 \to 4$. Ce *trail* couvre 5 sommets distincts. Cependant, ce *trail* n'est pas (D, G)-consistant parce que le sous-graphe de G induit par l'ensemble de sommets $\{7, 3, 1, 2, 4\}$ n'est pas connexe.

4.3 Modèles de programmation linéaire en nombres entiers

Dans la section précédente, nous avons proposé deux modèles de programmation linéaire en nombres entiers pour résoudre le problème One-to-One Skewgram. Pour rappel, ceux-ci se décomposaient chacun en deux ensembles de contraintes, l'un modélisant le chemin à chercher dans D, l'autre vérifiant la connexité dans G. Le premier ensemble de contraintes est le même dans les deux modèles. C'est donc le second ensemble de contraintes qui les différencie.

Nous cherchons maintenant un *trail*. Nous proposons donc de modifier le premier ensemble de contraintes. Le second, quelle que soit sa variante, n'a pas besoin d'être modifié. Nous aboutissons donc in fine à deux modèles, se différenciant dans la manière dont la connexité dans G est vérifié.

4.3.1 Trail dans D : Contraintes de trail communes aux deux modèles

Pour le graphe D, nous considérons dans les deux formulations deux sommets fictifs : un sommet source s et un sommet terminal t. Le sommet s est relié par des arcs sortants à tous les autres sommets dans D. Le sommet t est relié par des arcs entrants à tous les autres sommets dans D. Par souci de simplicité, nous définissons $A^+ = A \cup \{(s,i),\ \forall\ i \in V\} \cup \{(i,t),\ \forall\ i \in V\}$. Soit $K = \{1,2,..,|A|\}$.

Les variables utilisées dans le premier ensemble de contraintes sont des variables binaires communes aux deux modèles et sont définies comme suit :

(a) $x_{ij}^k = 1$ si et seulement si l'arc $(i,j) \in A^+$ est présent dans le *trail* cherché avec $k \in K$, l'ordre d'apparition de cet arc dans le *trail* cherché ;
(b) $z_i = 1$ si et seulement si le sommet $i \in V$ est présent dans le *trail* cherché.

Par exemple, le *trail* $((0,1)(1,2),(2,3),(3,1),(1,9),(9,10))$ correspond à la solution $x^1_{(0,1)} = 1$, $x^2_{(1,2)} = 1$, $x^3_{(2,3)} = 1$, $x^4_{(3,1)} = 1$, $x^5_{(1,9)} = 1$, $x^6_{(9,10)} = 1$ et $z_0 = z_1 = z_2 = z_3 = z_9 = z_{10} = 1$. Toutes les autres variables sont nulles. Les sommets 0 et 10 sont les deux sommets fictifs. Nous présentons ce *trail* avec l'ordre des arcs dans la Figure 4.2.

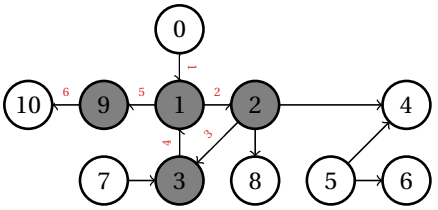


FIG. 4.2 : Exemple. Le *trail* $1 \to 2 \to 3 \to 1 \to 9$ couvre quatre sommets distincts. Les sommets 0 et 10 sont les deux sommets fictifs. Chaque arc est étiqueté par son ordre dans le *trail*.

Comme nous recherchons le *trail* de couverture maximale (D,G)-consistant, l'objectif est de maximiser le nombre de sommets utilisés dans la solution (c'est-à-dire le nombre de variables z_i fixées à 1),

$$max \sum_{i \in V} z_i \tag{4.1}$$

La première contrainte limite la sortie de la source s à un seul arc,

$$\sum_{(s,i) \in A} x^1_{si} = 1 \tag{4.2}$$

La seconde contrainte s'assure que l'on ne passe pas deux fois par un même arc.

$$\sum_{k \in K} x^k_{ij} \leq 1 \quad \forall (i,j) \in A \tag{4.3}$$

La troisième contrainte est la contrainte de conservation du flot; si on arrive au sommet l par l'arc d'ordre k, on sort de ce sommet par l'arc d'ordre $k+1$,

$$\sum_{i/(i,l) \in A} x^k_{il} = \sum_{j/(l,j) \in A} x^{k+1}_{lj} \quad \forall l \in V \quad \forall k \in K \tag{4.4}$$

Les deux dernières contraintes définissent la relation entre les variables z_i et x^k_{ij}. Le sommet $i \in V$ est dans le *trail* ($z_i = 1$) si et seulement si un arc $(i,j) \in A^+$ d'ordre k part de ce sommet ($x^k_{ij} = 1$),

$$z_i \leq \sum_{k \in K} \sum_{j=1}^{n} x^k_{ij} \quad \forall (i,j) \in A \tag{4.5}$$

$$z_i \geq x_{ij}^k \quad \forall (i,j) \in A \quad \forall k \in K \tag{4.6}$$

4.3.2 Connexité dans G : Contraintes de connexité

Pour assurer la connexité de l'ensemble des sommets sélectionnés dans le graphe non-orienté G, nous avons utilisé les deux formulations présentées dans le chapitre précédent. Pour rappel, ces deux formulations sont des adaptations de deux formulations du problème de l'arbre couvrant de poids minimal [Martin, 1991, Abdelmaguid, 2018].

Ces formulations sont détaillées dans la Section 3.5.

4.4 Modèle de programmation par contraintes

Nous présentons ci-dessous notre modèle de programmation par contraintes.

Pour rappel, nous avons $D = (V, A)$ un graphe orienté, $G = (V, E)$ un graphe non-orienté et $N = |V|$ le nombre de sommets. Nous ajoutons deux sommets fictifs s et t dans D. Nous ajoutons également un arc de t à s, exactement comme dans la section précédente. Ces deux sommets doivent être présents dans le *trail*. Soit $\varphi = (V \cup \{s,t\}, A \cup \{(t,s)\} \cup \{(s,i), i \in V\} \cup \{(i,t), i \in V\})$. Les variables de décision sont définies comme suit :

- $d \in [(\{s,t\}, \{(t,s)\}), \phi]$, une variable de type graphe orienté, représentant le *trail* retenu.
- $g \in [\emptyset, G]$: un sous-graphe connexe de G contenant les sommets retenus.
- $z_i \in \{0,1\}$: n variables booléennes indiquant si, $i \in V$ appartient au *trail* retenu.
- $deg_i \in \{0,\ldots,N\}$: indique le dégrée de $i \in V \cup \{s,t\}$ dans le *trail* retenu.
- $C \in \{2,\ldots,N\}$: la couverture du *trail* retenu.

La maximisation de la couverture du *trail* conduit à l'objectif suivant.

$$maximize(C)$$

Les contraintes sont formulées comme suit :

- Les deux premières contraintes impose la connexité de d et g.

$$connected(d) \tag{4.7}$$

$$connected(g) \tag{4.8}$$

- Ces deux contraintes assurent que les sommets retenus dans D et G sont les même.

$$nodesChanneling(d, \{z_i, i \in V\}) \tag{4.9}$$

$$nodesChanneling(g, \{z_i, i \in V\}) \tag{4.10}$$

- Ces deux contraintes imposent que les demi-degrés intérieur et extérieur de chaque sommet soient égaux. Ces contraintes peuvent être vues comme des contraintes de conservation de flot. Notez que l'on n'encode ainsi pas directement l'ordre dans lequel les arcs sont empruntés (mais une procédure de post-traitement polynomial peut calculer un tel ordre si nécessaire).

$$inDegrees(d, \{deg_i, i \in V\}) \tag{4.11}$$

$$outDegrees(d, \{deg_i, i \in V\}) \tag{4.12}$$

– La couverture du *trail* est égale au nombre de sommets retenus.

$$C = \sum_{i=1}^{n} z_i \tag{4.13}$$

4.4.1 Les stratégies de branchement

Au niveau de la phase de résolution d'un problème, le choix de la variable de branchement et de sa valeur est une étape très sensible car il a un impact très important sur la forme de l'arbre de recherche généré. Plus l'arbre est petit, meilleure est la performance. En programmation par contraintes, il existe plusieurs stratégies de branchement qui peuvent influencer la taille de l'arbre [Malapert, 2011]. Ces stratégies de branchement dépendent de :

– l'ordre dans lequel les variables sont choisies pour le branchement,
– l'ordre dans lequel les valeurs sont choisies pour l'exploration.

Dans ce qui suit, nous présentons quelques stratégies de branchement basées soit sur la sélection des variables, soit sur la sélection des valeurs. Ces stratégies sont présentées en détail dans [Malapert, 2011].

Parmi les stratégies les plus couramment utilisées par les solveurs de contraintes pour la sélection des variables, on cite :

– *Min-Domain*, qui sélectionne d'abord la variable non instanciée qui a le plus petit domaine de valeurs,
– *Random*, sélectionne aléatoirement une variable non instanciée,
– *Impact*, qui sélectionne la variable qui a le plus d'influence sur la réduction de l'espace de recherche,
– *Degree*, qui ordonne les variables selon leur degrés décroissants dans le graphe de contraintes [Dechter and Meiri, 1989].

Puisque nous utilisons des variable de type graphe, nous citons diverses heuristiques générales à ce type de variable [Fages et al., 2016].

– *LEXICO*, qui sélectionne la première arête non fixée.
– *MIN INF DEG (resp. MAX INF DEG)*, qui sélectionne une arête pour laquelle la somme des degrés des extrémités dans la borne inférieure du graphe est minimale (*resp.* maximale).
– *MIN SUP DEG (resp. MAX SUP DEG)*, qui sélectionne une arête pour laquelle la somme des degrés des extrémités dans la borne supérieure du graphe est minimale (*resp.* maximale).
– *MIN DELTA DEG (resp. MAX DELTA DEG)*, qui sélectionne une arête pour laquelle la somme des degrés des extrémités dans la borne supérieure du graphe moins la somme des degrés dans la borne inférieure du graphe est minimale (*resp.* maximale).

Une fois la variable sélectionnée, il faut brancher, naturellement, mais également choisir dans quel ordre les différentes sous-branches vont être parcourues. Parmi les stratégies basées sur la sélection de valeurs, nous citons :

– *minVal*, qui sélectionne la plus petite valeur du domaine,
– *maxVal*, qui sélectionne la plus grande,
– *randVal*, qui sélectionne aléatoirement une valeur.

Plusieurs stratégies de branchement ont été testés pour améliorer les performances en pratique de notre modèle. Nous avons ensuite développé une stratégie plus adaptée à notre problème. Nous appelons cette stratégie *walk-and-cover*. Cette derniere fonctionne comme suit :

1. sélectionner le premier sommet i non-équilibré (au sens des demi-degrés intérieur et extérieur),
2. brancher en priorité sur des arcs reliant i à des sommets successeurs non couvert,
3. brancher ensuite en priorité sur les arcs reliant i aux sommets couverts ayant beaucoup de successeurs.

Le point 1 permet en réalité de construire le *trail* en partant du sommet source s, puis en avançant de

sommet en sommet (*walk*). Il permet également de construire des solutions qui respectent naturellement la contrainte d'équilibre entre les demi-degrés intérieur et extérieur de chaque sommet.

Le point 2 permet de faire monter très rapidement la couverture du *trail* construit (*cover*).

Le point 3 permet, enfin, d'atteindre en priorité des sommets ayant de plus forte chance d'être connectés à des sommets pas encore couverts.

4.5 Résultats expérimentaux

Comme dans le chapitre précédent, nous allons montrer l'efficacité des méthodes proposées en termes de temps d'exécution et de qualité, en les appliquant sur des graphes aléatoires de type Erdos–Renyi et scale-free.

Enfin, nous appliquons également les méthodes proposées à des réseaux biologiques dans le contexte du réseau métabolique par rapport au génome.

Nous avons utilisé CPLEX 12.8.0 avec les paramètres par défaut pour l'implémentation des modèles linéaires. Nous avons utilisé Choco pour l'implémentation du modèle de programmation par contraintes avec les paramètres par défaut. Le temps d'exécution a été limité à 1300 s par instance pour les instances générées aléatoirement et à 300 s pour les données biologiques.

Dans tous les tableaux qui comparent les résultats pour des instances générées de façon aléatoire, les résultats de chaque méthode sont affichés dans deux colonnes : une colonne contient les temps d'exécution, et l'autre colonne contient les couvertures de solution. La couverture de la solution est en gras si l'optimalité de la solution été prouvée. Dans tous les tableaux de résultats pour des données biologiques, les résultats de chaque méthode sont présentées dans deux colonnes : une colonne pour le temps d'exécution, et l'autre colonne contient le nombre d'instances où une solution optimale a été trouvée. Le temps est présenté en secondes.

4.5.1 Graphes d'Erdos–Renyi

Pour comparer les résultats des quatre méthodes (HNet, ILP1, ILP2 et PPC), nous générons cinq instances pour chaque tuple (n, $p1$, $p2$), où

(a) $n \in \{30, 60, 110\}$ est le nombre de sommets ;
(b) $p1 \in \{0.05, 0.10, 0.20\}$ est la probabilité d'avoir un arc entre deux sommets pour D ;
(c) $p2 \in \{0.05, 0.10, 0.20\}$ est la probabilité d'avoir une arête entre deux sommets pour G.

Nous nous limitons dans ces expérimentations à présenter les instances les plus difficiles à résoudre à $p1 = 0.05$ et $p2 = 0.05$.

Le tableau 4.1 compare les performances des quatre méthodes proposées pour résoudre le problème. Dans le premier cas, $n = 30$, $p1 = 0.05$, et $p2 = 0.05$, et chaque méthode trouve la solution optimale. PPC et HNET sont similaires en termes de qualité et de temps d'exécution, alors que ILP1 et ILP2 nécessitent un temps d'exécution plus long. Dans le deuxième cas, $n = 60$, $p1 = 0.05$, et $p2 = 0.05$, et seulement ILP1 trouve la solution optimale. Dans le troisième cas, $n = 110$, $p1 = 0,05$, et $p2 = 0,05$, les quatre méthodes n'arrivent plus à trouver des solutions.

TAB. 4.1 : Instances de type Erdos–Renyi : $n \in \{30, 60, 110\}$, $p1 = 0.05$, et $p2 = 0.05$. La couverture de solution en gras indique que la solution est optimale.

			HNET		ILP1			ILP2			PPC	
Instance	$\|V\|$	$\|A\|$	$T(s)$	$\|Path\|$	$T(s)$	$\|Path\|$	Gap (%)	$T(s)$	$\|Path\|$	Gap (%)	$T(s)$	$\|Path\|$
1	30	41	2.48	**12**	2,94	**12**	0	5.62	**12**	0	0.06	**12**
2	30	37	0.11	**2**	1.30	**2**	0	0.63	**2**	0	0.05	**2**
3	30	43	0.93	**3**	1.35	**3**	0	1.93	**3**	0	0.05	**3**
4	30	49	0.18	**0**	2.02	**0**	0	8.99	**0**	0	0.06	**0**
5	30	43	0.35	**2**	1.48	**2**	0	0.64	**2**	0	0.06	**2**
6	60	182	1300	—	334.21	**54**	0	1300	54	0.02	1300	—
7	60	174	1300	—	266.19	**50**	0	1300	50	0.02	1300	—
8	60	180	1300	—	374.37	**52**	0	1300	38	0.42	1300	—
9	60	200	1300	—	953.71	**52**	0	1300	4	12.5	1300	—
10	60	192	1300	—	108.24	**53**	0	1300	53	0.01	1300	—
11	110	566	1300	—	1300	—	54	1300	—	—	1300	—
12	110	567	1300	—	1300	—	54	1300	—	—	1300	—
13	110	615	1300	—	1300	—	54.5	1300	—	—	1300	—
14	110	588	1300	—	1300	—	54.5	1300	—	—	1300	—
15	110	627	1300	—	1300	—	54.5	1300	—	—	1300	—

Dans cet ensemble de résultats numériques (graphes d'Erdos-Renyi), le nombre total d'instances est 15. En termes de qualité, HNet, PPC et ILP2 trouvent chacun 33 % des solutions optimales, alors ILP1 trouvent respectivement 66 % des solutions optimales. Pour le temps d'exécution moyen, HNet, PPC et ILP2 nécessitent 866 s pour chacun alors que ILP1 nécessite 569 s. Notez que HNet, PPC et ILP2 échouent pour $n > 10$, ILP1 échoue pour $\geqslant 110$.

Ces résultats confirment que ILP2 s'exécute plus rapidement et fournit solution de meilleure qualité que PPC et HNet, et que ILP1 s'exécute plus vite et fournit une solution de meilleure qualité que ILP2.

4.5.2 Graphes de scale-free

Pour comparer les résultats des quatre méthodes (HNet, ILP1, ILP2 et PPC), nous avons généré cinq instances pour chaque valeur de n, où
$n \in \{10, 60, 110\}$ est le nombre de sommets.

Le tableau 4.2 compare les résultats des quatre méthodes proposées pour résoudre le problème. Dans le premier cas, $n = 10$ et toutes les méthode trouvent la solution optimale sauf HNET. ILP1, ILP2 et PPC sont similaires en termes de qualité de solution et de temps d'exécution, alors que HNET échoue pour trouver une solution. Dans le deuxième cas, $n = 60$ et ILP1, ILP2 et PPC trouvent la solution optimale. Dans le troisième ($n = 110$) cas, seul PPC trouve la solution optimale. La PPC est globalement

TAB. 4.2 : Instances de type scale–free : $n \in \{10, 60, 110\}$. La couverture de solution en gras indique que la solution est optimale.

			HNET		ILP1			ILP2			PPC	
Instance	$\|V\|$	$\|A\|$	$T(s)$	$\|Path\|$	$T(s)$	$\|Path\|$	Gap (%)	$T(s)$	$\|Path\|$	Gap (%)	$T(s)$	$\|Path\|$
1	10	20	1300	—	0.28	**10**	0	0.30	**10**	0	0.034	**10**
2	10	24	1300	—	0.39	**10**	0	0.16	**10**	0	0.024	**10**
3	10	24	1300	—	0.41	**10**	0	0.15	**10**	0	0.008	**10**
4	10	22	1300	—	0.26	**10**	0	0.15	**10**	0	0.025	**10**
5	10	20	1300	—	0.40	**10**	0	0.21	**10**	0	0.008	**10**
6	60	178	1300	—	198.59	**60**	0	354.19	**60**	0	0.13	**60**
7	60	158	1300	—	212.25	**60**	0	121.28	**60**	0	0.09	**60**
8	60	184	1300	—	208.39	**60**	0	95.7	**60**	0	0.09	**60**
9	60	148	1300	—	107.81	**60**	0	43.2	**60**	0	0.10	**60**
10	60	166	1300	—	122.88	**60**	0	29.90	**60**	0	0.09	**60**
11	110	328	1300	—	1300	—	54.5	1300	—	—	0.26	**110**
12	110	310	1300	—	1300	—	54.5	1300	—	—	0.26	**110**
13	110	338	1300	—	1300	2	36	1300	—	—	0.26	**110**
14	110	358	1300	—	1300	—	54.5	1300	—	—	0.33	**110**
15	110	318	1300	—	1300	—	54.5	1300	—	—	0.35	**110**

beaucoup plus rapide.

Dans cet ensemble de résultats numériques (graphes de scale-free), le nombre total d'instances est 15. En termes de qualité et temps d'exécution la PPC est la meilleure.

Nous avons remarqué que pour ce type de graphe, l'ensemble des sommets d'un graphe forme une solution. Ceci est due au fait que les réseaux scale-free sont caractérisés par la présence de hubs, ou de quelques noeuds fortement connectés aux autres noeuds du réseau et puisque nous cherchons des *trails* (on peut passer plusieurs fois par le même sommet). Il est alors possible que l'ensemble de sommets d'un graphe forme une solution.

4.5.3 Données biologiques

Comme dans la section 4.5.3, Nous appliquons les méthodes proposées pour extraire automatiquement d'une voie métabolique une chaîne de réactions qui sont catalysées par les produits de gènes voisins dans le génome.

Comme dans une approche de la littérature basée sur les graphes pour l'intégration de données biologiques hétérogènes dans un autre contexte [Boyer et al., 2005], une étape de prétraitement a été ajoutée à afin de tenir compte des réactions et/ou des gènes non contigus. L'étape de prétraitement consiste à modifier les graphes d'entrée en ajoutant des arcs (respectivement des arêtes) entre les sommets sépa-

TAB. 4.3 : Instances biologiques : Voie métabolique versus génome

5378 instances de 50 espèces

	HNet		ILP1		ILP2		PPC2	
Number of instances	$T(s)$	Optimal	$T(s)$	Optimal	$T(s)$	Optimal	$T(s)$	Optimal
5378	41165	5261	.	.	128867	5335	33774	5360

rés par au plus δ_D autres réactions (respectivement δ_G autres gènes). Les valeurs de $\delta_D \in \{0,1,2,3\}$ et/ou $\delta_G \in \{0,1,2,3\}$ doivent être fixées assez bas (par exemple, au plus 3) pour garantir que les *trails* produits sont pertinents d'un point de vue biologique.

Nous avons récupéré les informations métaboliques et génomiques de 50 espèces dans la base de données KEGG. Comme dans le chapitre précédent, nous avons considéré toutes les voies de chaque espèce. Cela donne un total de 5378 instances. Le temps limite pour chaque instance est de 300 secondes.

Les tableau 5.4 présente les résultats des quatre méthodes proposées pour résoudre le problème. En ce qui concerne la qualité des résultats, PPC, HNet, ILP1 et ILP2 trouvent 99.66, 97.77, ... et 99.20% des solutions optimales, respectivement. En termes de temps d'exécution, HNet est plus rapide que ILP2, PPC est plus rapide que HNet.

Pour rappel, les instances réelles ressemblent davantage à des graphes scale-free qu'à des graphes Erdos-Renyi. Nous pouvons donc dire que nos méthodes sont adaptées aux instances réelles ; ce que nos expérimentations ont d'ailleurs confirmé.

4.5.3.1 Exemple de trail

Comme dans le chapitre précédent, nous présentons dans cette section des figures illustrant le regroupement des *trails* (une chaines de réactions catalysées par le produits des gènes voisines).

La Figure 4.3 montre une chaine de réactions de l'espèce d'*Bifidobacterium breve* dans la voie de *Arginine biosynthesis*. La chaine de réactions $R02282 \rightarrow R02649 \rightarrow R03443 \rightarrow R02283 \rightarrow R00259 \rightarrow R01954 \rightarrow R01086$ est catalysé par le produits des gènes $bbv : HMPREF9228\_1182$, $bbv : HMPREF9228\_1181$, $bbv : HMPREF9228\_1183$, $bbv : HMPREF9228\_1180$, $\{bbv : HMPREF9228\_0222, bbv : HMPREF9228\_1182\}$, $bbv : HMPREF9228\_1179$, $bbv : HMPREF9228\_1177$, $bbv : HMPREF9228\_1170$.

4.6 Conservation du trail

Dans les sections précédentes, des méthodes ont été présentées pour trouver les *trails* des réactions qui sont catalysées par les produits de gènes voisines pour une espèce donnée. La présente section illustre comment ces *trails* peuvent être exploités pour étudier la conservation des voies métaboliques et génomiques entre différentes espèces.

Dans cette section, nous nous concentrons sur deux espèces, à savoir *Bifidobacterium breve* et *Bacteroides fragilis*, et montrons que les chemins extraits dans l'espèce *Bifidobacterium breve* peuvent être totalement conservés (trait commun), partiellement conservés (peut-être la résultat d'une évolution) ou non conservés (différence) dans l'espèce *Bifidobacterium breve*. Un *trail* est conservé entre deux espèces s'il existe et s'il est constant dans les deux espèces. Un *trail* est partiellement conservé si seulement un sous-*trail* DG-consistant de celui-ci, existe dans la seconde espèce.

Dans la voie *Arginine biosynthesis* de l'espèce *Bifidobacterium breve*, PPC a trouvé le *trail* des réactions $R02282 \rightarrow R02649 \rightarrow R03443 \rightarrow R02283 \rightarrow R00259 \rightarrow R01954 \rightarrow R01086$.]La chaine $R02282 \rightarrow R02649 \rightarrow R03443 \rightarrow R02283 \rightarrow R00259 \rightarrow R01954 \rightarrow R01086$ est catalysé par le produits des gènes $bbv :$

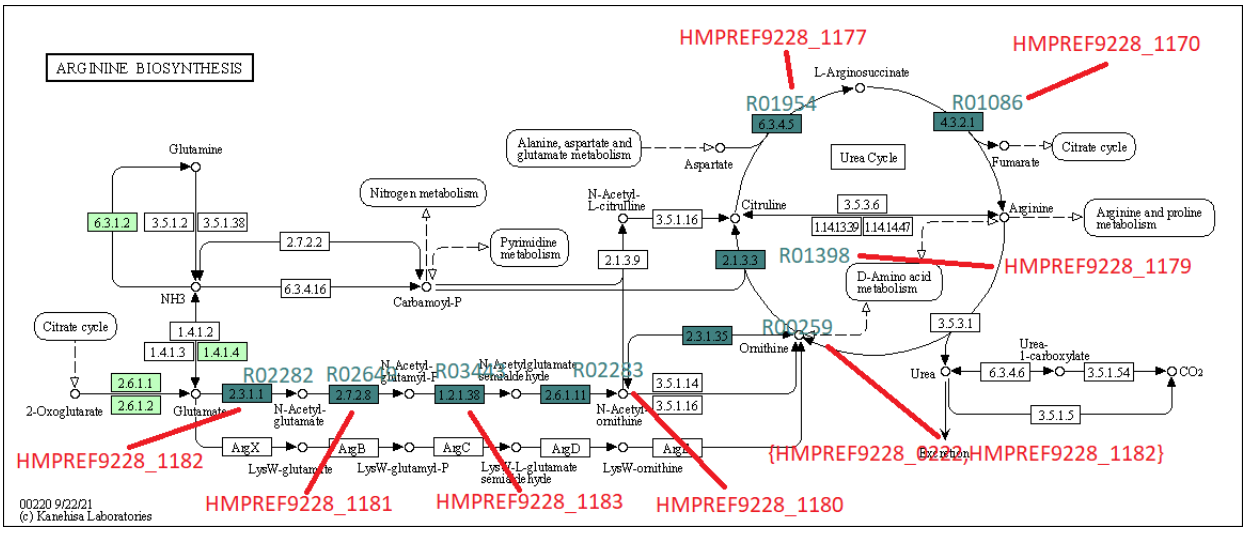


FIG. 4.3 : Une chaine de réactions de l'espèce d'*Bifidobacterium breve* dans la voie de *Arginine biosynthesis* catalysée par le produits des gènes voisins. La chaine $R02282 \rightarrow R02649 \rightarrow R03443 \rightarrow R02283 \rightarrow R00259 \rightarrow R01954 \rightarrow R01086$ est catalysé par le produits des gènes $bbv : HMPREF9228\_1182, bbv : HMPREF9228\_1181, bbv : HMPREF9228\_1183, bbv : HMPREF9228\_1180, \{bbv : HMPREF9228\_0222, bbv : HMPREF9228\_1182\}, bbv : HMPREF9228\_1179, bbv : HMPREF9228\_1177, bbv : HMPREF9228\_1170$. On voit ici un *trail* constituée par les réactions avec un couleur de fond bleu. Les réactions du *trail* sont étiquetées avec les identifiants de réaction KEGG correspondants (*R numbers*) et avec les identifiants des gènes impliqués dans les réactions. Les gènes avec des identificateurs bleu sont voisins.

$HMPREF9228\_1182, bbv : HMPREF9228\_1181, bbv : HMPREF9228\_1183, bbv : HMPREF9228\_1180, \{bbv : HMPREF9228\_0222, bbv : HMPREF9228\_1182\}, bbv : HMPREF9228\_1179, bbv : HMPREF9228\_1177, bbv : HMPREF9228\_1170$. Dans l'espèce *Bacteroides fragilis*, PPC a trouvé un sous *trail* de réactions $R03443 \rightarrow R02283$, catalysé par les produits des gènes voisins $\{bfr : BF0532, bfr : BF0531\}$. Nous pouvons alors conclure que ce *trail* est partiellement conservé entre les deux espèces (*cf*. figure 4.4).

4.7 Conclusion

Dans ce chapitre, nous avons présentés une étude d'un problème d'optimisation combinatoire appelé "SkewGraM". La complexité de ce problème est $\mathcal{NP}$-difficile au sens fort. Ce problème a des applications dans l'analyse des réseaux biologiques et des applications prévisibles dans l'analyse des réseaux sociaux, d'information et de communication.

Pour résoudre ce problème, nous avons proposé deux modèles linéaires et un modèle de programmation par contraintes et avons comparé les résultats de ces méthodes proposées avec une méthode de la littérature appelée HNet [Zaharia et al., 2019] en termes de temps d'exécution et de qualité en les appliquant sur des graphes aléatoires et des données biologiques.

Les résultats présentés améliorent nettement les meilleurs résultats de la littérature [Zaharia et al., 2019].

Dans le chapitre qui suit, nous présenterons l'interprétation biologique de ces résultats.

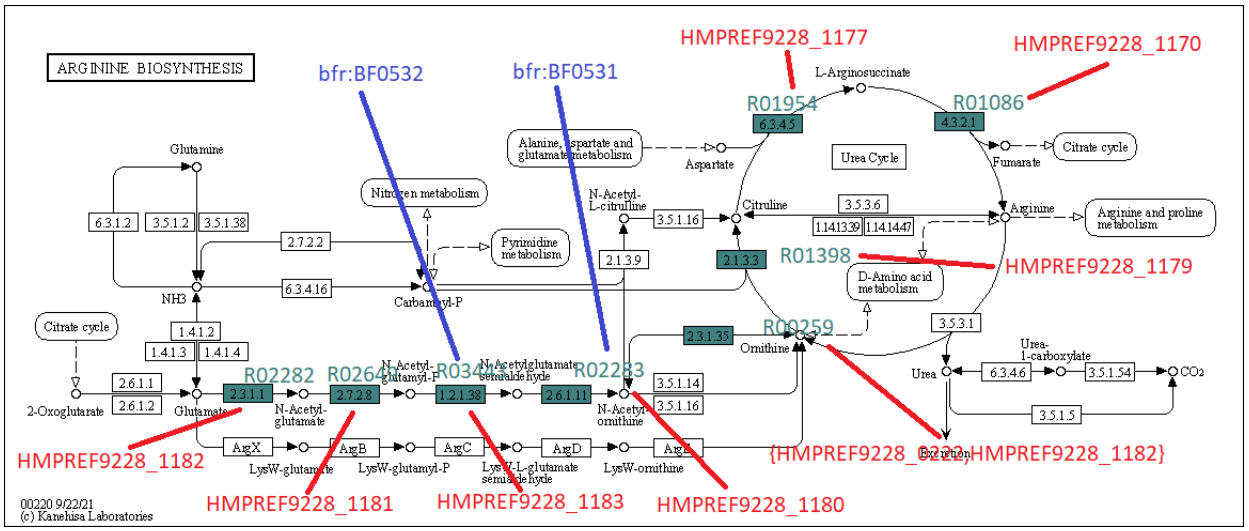


FIG. 4.4 : *Trail* partiellement conservé (peut-être la résultat d'une évolution qui est un processus par lequel les espèces se transforment)

Chapitre 5

Conservations des motifs métaboliques et génomiques

Contents

5.1 Introduction

Les chapitres précédents traitent de la recherche d'une solution optimale aux problèmes One-to-One SkewGraM et SkewGraM. Des méthodes qui identifient des chemins ou des *trails* de réactions catalysées par les produits de gènes voisins pour une espèce donnée ont été proposées. Dans ce chapitre, nous cherchons une *famille* de *trails*, souvent préférée aux solutions uniques par les biologistes, puis nous montrons comment ces motifs métaboliques et génomiques propres à chaque espèce peuvent être exploités afin de détecter la conservation dans des espèces différentes. Nous présentons d'abord les méthodes développées pour chercher une *famille* de *trails* (plutôt qu'une solution unique), puis nous introduisons le concept de conservation des *trails* présentés dans les travaux de Zaharia *et al.* [Zaharia et al., 2019]. Nous verrons ainsi comment nos familles de *trails* peuvent être exploitées d'un point de vue biologique. Nous verrons notamment qu'il est possible d'étudier la conservation des motifs métaboliques et génomiques selon deux points de vue. Premièrement la conservation des *trails* par réactions qui se concentre sur l'aspect métabolique. En second, la conservation des *trails* par gènes qui se concentre sur l'aspect génomique. Ainsi, dans ce chapitre, nous proposerons tout d'abord des techniques adaptées pour la recherche d'une *famille* de *trails* plutôt que d'une solution unique. Nous terminons ce chapitre par une discussion générale sur la conservation des *trails* entre différentes espèces.

5.2 Recherche de famille de trails

5.2.1 Méthodes pour la recherche d'un trail passant un arc

Les méthodes proposées dans les chapitres précédents permettent de trouver un *trail* de couverture maximale passant par un arc donné. En passant en revue chaque arc de D (en résolvant m fois le problème, une fois par arc), on peut facilement générer une famille de *trail*. Nous discuterons les résultats obtenus avec cette approche, pour chaque méthode (modèles linéaires et de programmation par contraintes) dans la section 5.2.3.

5.2.2 Méthode pour la recherche de tous les trails distincts

Dans la présente section, nous adaptons le modèle de programmation par contraintes proposé dans la section 4.4, afin de trouver toutes les solutions distinctes. Par solutions distinctes, nous entendons de couverture différente. Plusieurs *trails* peuvent en effet avoir une couverture similaire, mais nous verrons plus tard que considérer plusieurs solutions de même couverture ne présente aucun intérêt pour l'interprétation biologique.

Rappelons pour commencer qu'on appelle *nogood*, tout sous-ensemble de décisions qui ne peut être étendu à aucune solution [Dechter, 1990]. Par exemple, tout ensemble d'affectations qui viole une contrainte est un *nogood*. De même, un *CSP* n'a pas de solution si et seulement si l'ensemble vide d'affectations est un *nogood*. Si une instanciation partielle recouvre un nogood, alors toutes les extensions de cette instanciation recouvriront ce *nogood*, et aucune ne pourra être étendue à une solution. Il convient alors de ne pas explorer le sous-arbre induit et de revenir en arrière pour poursuivre la recherche. Les *nogoods* sont découverts au fur et à mesure de l'exploration de l'espace de recherche, lorsque des échecs sont rencontrés. Il existe en effet des algorithmes polynomiaux capables d'analyser les raisons d'un échec, et d'en déduire des *nogoods* minimum au sens de l'inclusion, voir par exemple [Junker, 2001]. Ces *nogoods* sont alors stockés dans une base de *nogoods*, qui agit comme une contrainte en filtrant des valeurs du domaine des variables présentes dans les *nogoods*. Les sous-espaces ne menant à aucune solution, lorsqu'ils sont détectés, sont donc mémorisés et évités en poursuivant l'exploration de l'espace de recherche.

Nous proposons d'exploiter ce mécanisme pour la recherche de *trails* distincts. Nous pouvons en effet, à chaque fois qu'un *trail* valide est trouvé, enregistrer en tant que *nogoods* l'ensemble des z_i fixés 0 (l'ensemble des sommets non retenus). Le solveur ne pourra ainsi plus trouver de solution dont la couverture est inclue ou égale à celle d'une solution précédemment trouvée. Il est néanmoins toujours possible que le solveur trouve plus tard une solution dont la couverture est un sur-ensemble de celle d'une solution trouvée précédemment. Bien que très efficace en pratique, cette technique permet donc pas totalement de se passer d'une procédure *ad-hoc* de suppression des solutions équivalentes au sens de la couverture.

5.2.3 Résultats expérimentaux

Comme dans le chapitre précédent, nous allons montrer l'efficacité des méthodes proposées en termes de temps d'exécution et de qualité, en les appliquant sur des données réelles.

L'objectif principal est d'étudier la relation entre le métabolisme et le génome. Nous avons choisi de nous concentrer sur la détection de réactions voisines catalysées par des produits de gènes voisins.

Pour des raison biologique, nous avons choisi de nous concentrer sur les procaryotes [Moreno-Hagelsieb and Santoyo, 2015]. Un ensemble de données de 50 espèces bactériennes couvrant les principaux phyla de l'arbre de vie bactérien a été choisi (*cf.* Tableau 5.1). L'ensemble des données est donc représentatif de l'ensemble du domaine bactérien.

Nous avons récupéré les informations métaboliques et génomiques de ces 50 espèces dans la base de données KEGG (l'encyclopédie des gènes et des génomes de Kyoto). Nous avons considéré toutes les voies de chaque espèce. Cela donne un total de 5378 instances.

Espèces	Souche	Classe	code KEGG
Escherichia coli	K-12 MG1655	γ-proteobacteria	eco
Yersinia pestis	CO92 (biovar Orientalis)	γ-proteobacteria	ype
Vibrio cholerae	O395	γ-proteobacteria	vco
Shewanella putrefaciens	CN-32	γ-proteobacteria	spc
Pseudomonas aeruginosa	PAO1	γ-proteobacteria	pae
Xylella fastidiosa	9a5c	γ-proteobacteria	xfa
Ralstonia solanacearum	GMI1000	β-proteobacteria	rso
Neisseria meningitidis	MC58 (serogroup B)	β-proteobacteria	nme
Acidithiobacillus ferrivorans		Acidithiobacillia	afi
Agrobacterium radiobacter		α-proteobacteria	ara
Rickettsia rickettsii	Iowa	α-proteobacteria	rrj
Geobacter sulfurreducens	PCA	δ-proteobacteria	gsu
Nitrospira defluvii		Nitrospira	nde
Acidobacterium capsulatum		Acidobacteriales	aca
Desulfurispirillum indicum		Chrysiogenetes	din
Fusobacterium nucleatum	subsp. nucleatum ATCC 25586	Fusobacteriia	fnu
Denitrovibrio acetiphilus		Deferribacteres	dap
Thermodesulfatator indicus		Thermodesulfobacteria	tid
Aquifex aeolicus		Aquificae	aae
Bacillus subtilis	subsp. subtilis 168	Bacilli	bsu
Listeria monocytogenes	EGD-e	Bacilli	lmo
Staphylococcus aureus	subsp. aureus N315 (MRSA/VSSA)	Bacilli	sau
Lactobacillus acidophilus	NCFM	Bacilli	lac
Streptococcus pneumoniae	ST556	Bacilli	snd
Clostridium perfringens	13	Clostridia	cpe
Mycoplasma pneumoniae	M129	Mollicutes	mpn
Synechocystis sp.	PCC 6803	Cyanobacteria (phylum)	syn
Prochlorococcus marinus	subsp. marinus CCMP1375	Cyanobacteria (phylum)	pma
Chloroflexus aurantiacus		Chloroflexia	cau
Bifidobacterium breve	ACS-071-V-Sch8b	Actinobacteria	bbv
Corynebacterium glutamicum	ATCC 13032 (Kyowa Hakko)	Actinobacteria	cgl
Mycobacterium tuberculosis	H37Rv	Actinobacteria	mtv
Streptomyces coelicolor		Actinobacteria	sco
Deinococcus radiodurans		Deinococci	dra
Thermus thermophilus	HB27	Thermi	tth
Fimbriimonas ginsengisoli		Fimbriimonadia	fgi
Acetomicrobium mobile		Synergistia	amo
Thermotoga maritima	MSB8	Thermotogae	tmm
Caldisericum exile		Caldisericia	cex
Dictyoglomus thermophilum		Dictyoglomia	dth
Fibrobacter succinogenes		Fibrobacteria	fsu
Gemmatimonas aurantiaca		Gemmatimonadetes	gau
Chlorobium phaeobacteroides	DSM 266	Chlorobia	cph
Bacteroides fragilis	YCH46	Bacteroidia	bfr
Rhodopirellula baltica		Planctomycetia	rba
Chlamydia pneumoniae	CWL029	Chlamydiia	cpn
Opitutus terrae		Opitutae	ote
Borrelia burgdorferi	N40	Spirochaetia	bbn
Elusimicrobium minutum		Elusimicrobia	emi
Helicobacter pylori	26695	ε-proteobacteria	heo

TAB. 5.1 : L'ensemble des données de 50 espèces bactériennes étudiées.

Comme dans une approche de la littérature basée sur les graphes pour l'intégration de données biologiques hétérogènes dans un autre contexte [Boyer et al., 2005], une étape de prétraitement a été ajoutée à afin de tenir compte des réactions et/ou des gènes non contigus. L'étape de prétraitement consiste à modifier les graphes d'entrée en ajoutant des arcs (respectivement des arêtes) entre les sommets séparés par au plus δ_D autres réactions (respectivement δ_G autres gènes). Les valeurs de $\delta_D \in \{0,1,2,3\}$ et/ou $\delta_D \in \{0,1,2,3\}$ doivent être fixées assez bas (par exemple, au plus 3) pour garantir que les *trails* produits sont pertinents d'un point de vue biologique.

Pour rappel, le génome est modélisé par un graphe non-orienté dont les sommets représentent les gènes et une arête sépare deux gènes adjacents. Nous avons utilisé CPLEX 12.8.0 avec les paramètres par défaut pour l'implémentation des modèles linéaires. Nous avons utilisé Choco pour l'implémentation du modèle de programmation par contraintes avec les paramètres par défaut. Le temps d'exécution à 300 s pour les données biologiques. Dans les tableaux de statuts des méthodes, la solution d'une méthode a l'un des statuts suivants :

- UNKOWN (TIME) : la méthode n'a pas trouvé une solution en temps limite.
- FEASIBLE : la méthode a trouvé une solution sans garantie de son optimalité.
- INFEASIBLE : la méthode a prouvé qu'il n'y a pas une solution (s et t ne sont pas dans G).
- OPTIMAL (OK) : la méthode a prouvé l'optimalité de la solution (temps<temps limite).
- INIT : échec d'initialisation.

Dans tous les tableaux de résultats pour des données biologiques, les résultats de chaque méthode sont présentées dans trois colonnes : une colonne pour le nombre de *trails*, une colonne pour la couverture moyenne des trails, et une colonne pour le temps d'exécution. Le temps est présenté en secondes. Pour rappel, la couverture d'un *trail* est le nombre de sommets distincts présents dans un *trail*.

À noter que pour les des méthodes de recherche d'un chemin, un pré-traitement pour rendre le graphe D acyclique est nécessaire. Ce pré-traitement a grand impact sur l'instance (n'est plus la même instance).

Le tableau 5.2 compare les performances des méthodes (arc par arc) proposées pour résoudre le problème. Nous prenons en considération tous les δ_D et δ_G. Le nombre d'échec d'initialisation est le même. Les nombre de fois où une méthode n'a pas trouvé une solution sont 0, 175, 450 et 1096 pour les méthodes PPC (chemins), PPC (trails), PLNE 2 et HNET respectivement.

TAB. 5.2 : Statut des méthodes de résolutions.

	TOUS LES deltaD ET deltaG				
	STATUT				
	INIT	**UNKOWN (TIME)**	**FEASIBLE**	**INFEASIBLE**	**OPTIMAL (OK)**
HNET (*)	11	1096	14	29475	26652
PLNE 2 (arc by arc) (*)	11	450	12	29537	27238
PPC (trails) (arc by arc) (*)	11	171	68	29500	27498
PPC (chemins) (arc by arc) (*)	11	0	24	33771	23442

Le tableau 5.3 présente les résultats des méthodes (arc par arc) proposées pour résoudre le problème. Le temps d'exécution est 110337, 254787, 299279 et 402482 second pour les méthodes PPC (chemins), PLNE 2, PPC (trails) et HNET respectivement. La couverture moyenne du trail est 2.58, 2.52, 2.49 et 2.15 pour les méthodes PPC (trails), PLNE 2, HNET et PPC (chemins) respectivement.

TAB. 5.3 : Résultats de comparaisons des méthodes de résolution.

	TOUS LES deltaD ET deltaG		
	RÉSULTATS		
	nb trails	**couverture moyenne**	**time**
HNET (*)	150	2.49	402482
PLNE 2 (arc by arc) (*)	148	2.52	254787
PPC (trails) (arc by arc) (*)	154	2.58	299279
PPC (chemins) (arc by arc) (*)	265	2.15	110337

Le tableau 5.4 compare les performances des méthodes proposées pour résoudre le problème. Nous prenons en considération le cas où $\delta_D = 3$ et $\delta_G = 3$. Le nombre d'échec d'initialisation est le même. Les nombre de fois ou une méthode n'a pas trouvé une solution sont 0, 0, 13, 16 et 40 pour les méthodes PPC (chemins) (all distinct), PPC (chemins) (arc by arc), PPC (trails) (arc by arc), PPC (trails) (all distinct) et PLNE 2 (arc by arc) (3,3) respectivement.

TAB. 5.4 : Statut des méthodes de résolutions.

	deltaD = deltaG = 3				
	STATUT				
	INIT	**UNKOWN (TIME)**	**FEASIBLE**	**INFEASIBLE**	**OPTIMAL (OK)**
PLNE 2 (arc by arc) (3,3)	3	40	2	1727	1806
PPC (trails) (arc by arc) (3,3)	3	13	7	1727	1828
PPC (chemins) (arc by arc) (3,3)	3	0	3	1974	1598
PPC (trails) (all distinct) (3,3)	3	16	1	36	3522
PPC (chemins) (all distinct) (3,3)	3	0	2	1958	1615

Le tableau 5.5 présente les résultats des méthodes proposées pour résoudre le problème. Le temps d'exécution est 7560, 10632, 23641, 26795 et 33774 secondes pour les méthodes PPC (chemins) (all distinct), PPC (chemins) (arc by arc), PLNE 2 (arc by arc), PPC (trails) (arc by arc) et PPC (trails) (all distinct) respectivement. La couverture moyenne du trail est 2.82, 2.66, 2.52, 2.39 et 2.39 pour les méthodes PPC (trails) (all distinct), PPC (trails) (arc by arc), PLNE 2 (arc by arc), PPC (chemins) (all distinct) (3,3) et PPC (chemins) (arc by arc) (3,3) respectivement.

TAB. 5.5 : Résultats de comparaisons des méthodes de résolution.

deltaD = deltaG = 3			
RÉSULTATS			
	nb trails	**couverture moyenne**	**time**
PLNE 2 (arc by arc) (3,3)	130	2.52	23641
PPC (trails) (arc by arc) (3,3)	137	2.66	26795
PPC (chemins) (arc by arc) (3,3)	164	2.19	10632
PPC (trails) (all distinct) (3,3)	136	2.82	33774
PPC (chemins) (all distinct) (3,3)	172	2.39	7560

Ces résultats montrent que les méthodes peuvent trouver plus ou moins de *trails* distincts, qui sont plus ou moins longs. Bien sûr, plus les *trails* sont longs, plus ils en incluent d'autres, et donc moins de *trails* trouvés au final. Ces deux métriques sont donc liées. Nous avons remarqué également que les méthodes prennent plus ou moins de temps. Il y a donc un compromis à trouver entre le temps de résolution, le nombre et la longueur des *trails*.

5.3 Conservation des trails

Dans cette section, nous allons commencer par décrire l'approche proposée par Zaharia *et al* dans [Zaharia et al., 2019] pour étudier la conservation des *trails* à l'échelle de plusieurs espèces. Cette approche comparative se propose d'examiner les *trails* d'une espèce de référence donnée en fonction de leurs conservations métabolique et génomique chez d'autres espèces. Elle permet, par exemple, d'extraire de potentiels liens de parenté entre les espèces étudiées (traits hérités et/ou différentiés).

Une voie métabolique est souvent présente dans un groupe d'espèces. Les réactions de cette voie métabolique peuvent apparaitre totalement ou partiellement dans une espèce donnée. Dans la base de données KEGG, les cartes de référence fournissent une vue globale du métabolisme en cumulant toutes les variations métaboliques connues pour chaque organisme séquencé. Les cartes des voies métaboliques pour une espèce donnée sont donc des sous-ensembles des cartes de référence, dans lesquelles seules les réactions connues pour être produites par l'espèce donnée sont marquées.

Nous prenons comme exemple une voie métabolique et les contextes génomiques pour trois espèces. L'approche est illustrée dans l'exemple de la Figure 5.1 :

- le *trail* $T = (r_1, r_2, r_3)$ a été identifié pour les espèces S_1 et S_2, T est catalysé par les produits des gènes voisines $\{A_1, B_1, C_1\}$ et $\{A_2, B_2, C_2\}$ dans S_1 et S_2 respectivement, $\{A_2, B_2, C_2\}$ sont voisines dans S_2 car le gène X_2 est omis d'une distance $\delta_G = 1$;
- le *trail* $T' = (r_1, r'_2, r_3)$ a été identifiée pour l'espèce S_3, ce *trail* est catalysés par les produits des gènes voisines $\{A_3, B_3, C_3\}$;
- la réaction r'_2 n'est pas présente dans les espèces S_1 et S_2 ;
- la réaction $r_2 \notin T'$ est présente dans l'espèce S_3.

Dans la Figure 5.1, les gènes sont représentés par des rectangles. Les gènes voisins sont sont reliés par des arêtes. Les réactions spécifiées au dessus des gènes sont les réactions catalysées par les produits des gènes en question, à l'exception du gène X_2 de l'espèce S_2 qui ne code pas d'une enzyme.

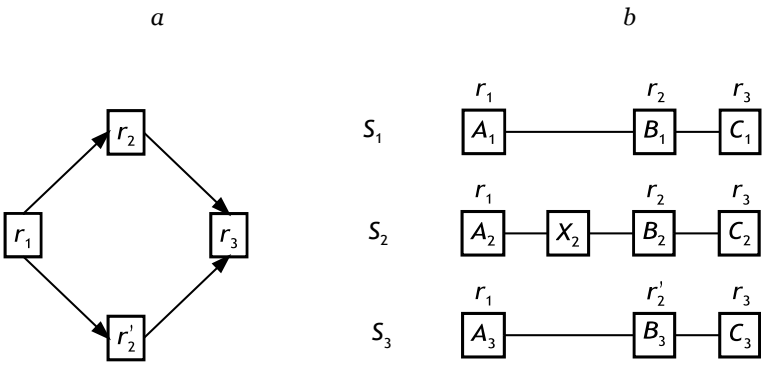


FIG. 5.1 : Une voie métabolique et le contexte génomique pour trois espèces. *(a)* Une voie métabolique de quatre réactions. *(b)* Contexte génomique pour trois espèces S_1, S_2 et S_3.

L'approche détermine :

a) si les réactions impliquées dans les *trails* de l'espèce de référence sont aussi catalysées par des produits de gènes voisins dans d'autres espèces,
b) si les gènes de l'espèce de référence impliqués dans ces *trails* donnés ont des gènes voisins fonctionnellement similaires dans d'autres espèces. Des gènes fonctionnellement gènes similaires codent des enzymes qui catalysent la même réaction.

Elle détermine *in fine* si un *trail* d'une espèce est partiellement ou entièrement conservé dans l'espèce de référence. Par exemple, le *trail* T est présent entièrement dans les espèces S_1 et S_2, alors que seules les réactions r_1 et $r3$ du *trail* T' sont présentes dans ces espèces. Cela permet d'étudier efficacement les variations entre les espèces à la fois au niveau métabolique et génomique.

5.3.1 Supports des trails

La conservation des *trails* traite les *trails* comme des ensembles de réactions, ce qui signifie que l'ordre des réactions n'est pas pris en compte et que le nombre d'occurrences d'une réaction dans un *trail* est ignoré. Par exemple, dans la Figure 5.2, les *trails* $T_1 = (1,2,3,1,9)$ et $T_2 = (2,3,1,9)$ ont le même ensemble de réactions, appelé support, $\{1,2,3,9\}$. Le support d'un *trail* T, noté $Sup(T)$, est l'ensemble des réactions qui le composent. Soit $pos(r,S)$, l'ensemble des positions des gènes impliqués dans une réaction r. Des gènes sur le même brin d'un chromosome donné sont considérés comme voisins s'ils sont séparés d'au plus δ_G autres gènes (si $\delta_G = 0$, deux gènes ne sont voisins que s'ils sont adjacents). Soit $genset(T,S)$, les positions des gènes impliqués dans $Sup(T)$, $genset(T,S) = \cup_{r\in T} pos(r,S)$.

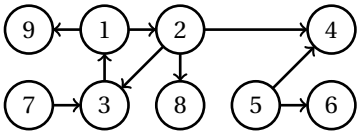


FIG. 5.2 : Exemple. Voie métabolique

Soit $P = \{1,\ldots,p\}$ un panel de p espèces à étudier. La conservation des *trails* nécessite la désignation d'une espèce de référence *ref* parmi les espèces de P. Les *trails* de l'espèce de référence obtenus via les méthodes présentées dans les chapitres précédents sont traités comme des ensembles de réactions afin d'étudier la conservation des motifs métaboliques et génomiques chez les espèces restantes dans P. On note $trails(S)$, l'ensemble des *trails* considérés dans l'étude d'une espèce S, et $Sup_{ref} = \cup_{T\in trails(ref)} Sup(T)$, l'union des supports de *trails* de l'espèce de référence. Comme dans [Boyer et al., 2005] et pour que la conservation des *trails* puisse tenir compte des variations génomiques entre les espèces, on considère dans cette étude que deux gènes d'une même espèce sont voisins s'ils sont séparés par au plus trois autres gènes sur le même brin du même chromosome.

La définition de motifs métaboliques et génomiques conservés s'adapte à de légères variations entre les espèces. L'une de ces variations est l'ordre des réactions entre les *trails*. Par exemple, si les *trails* (r_1, r_2) et (r_2, r_1) sont identifiés pour deux espèces différentes pour une voie métabolique, ces *trails* constituent un motif conservé pour les deux espèces. De façon plus générale, deux *trails* ayant un support identique sont considérés comme conservés, indépendamment de l'ordre des réactions et la présence ou non de réactions répétées. Un autre exemple de variation qui ne devrait pas empêcher l'identification de motifs conservés est lié aux réactions (ou gènes) qui sont présentes dans les *trails* de certaines, mais pas toutes les espèces.

Dans ce qui suit, nous présentons la conservation des *trails* par réactions et par gènes. Dans les deux cas, il s'agit d'identifier des motifs de l'espèce de référence dans les autres espèces.

– La conservation des *trails* par réactions se concentre davantage sur les motifs métaboliques conservés plutôt que sur les motifs génomiques.

– La conservation des *trails* par gènes se concentre davantage sur les motifs génomique conservés plutôt que sur les motifs métaboliques.

Supposons que le *trail* $T = (r_2, r_3, r_1, r_9)$, catalysée par les produits des gènes voisins $\{T, U, V, W\}$ et a été identifiée par les méthode présentées dans les chapitres précédents dans la voie de la Figure 5.3 pour une espèce de référence *ref*. Les contextes génomiques de l'espèce *ref* et d'une autre espèce S sont présentés dans la Figure 5.3. Dans cette dernière, les gènes appartenant au même brin chromosomique sont représentés par des rectangles. Les gènes sont représentés par des rectangles. Les gènes voisins sont sont reliés par des arêtes. Les réactions spécifiées au dessus des gènes sont les réactions catalysées par les produits des gènes en question, à l'exception du gène V_1 de l'espèce S qui code pas d'une enzyme. Les réactions appartiennent à la voie de la Figure 5.2 représente un ensemble de réactions de *ref* trouvées par les méthodes présentées dans les chapitres précédents. $R' = \{r_2, r_9\}$ est un *trail* catalysé par les produits des gènes voisins $\{U_1, V_1\}$. $\{U_1, V_1\}$ sont voisins dans S car le gène V_1 est omis d'une distance $\delta_G = 1$. Ce *trail* désigne un sous-ensemble maximal de R tel que les gènes de S impliqués dans les réactions de R' (en gras) soient voisins.

Les deux méthodes seront illustrées sur cet exemple.

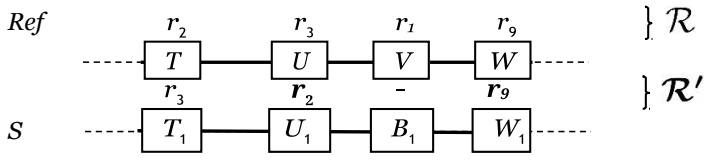


FIG. 5.3 : Voisinage des gènes pour les espèces *ref* et S.

5.3.2 Conservation des trails par réactions

La conservation des *trails* par réactions d'une espèce de référence *ref* consiste à construire un tableau dont les lignes représentent les réactions du support de chaque *trail* de ref et les colonnes représentant les autres espèces dans P. Les colonnes sont ordonnées en fonction de la distance évolutive par rapport à *ref*, de sorte que les espèces phylogénétiquement plus proches de *ref* ont des indices de colonne inférieures à celles qui sont phylogénétiquement éloignées de ref. Notez qu'une réaction donnée de l'espèce ref apparaît plusieurs fois dans ce tableau si elle apparait dans plusieurs *trails* de ref. Le tableau reflète les motifs métaboliques conservés entre l'espèce de référence et les autres espèces au travers des trois symboles qui peuvent être attribués à chaque cellule. Ces symboles sont :

– (x), réactions catalysées par les produits de gènes voisines,
– (.), réactions catalysées par les produits de gènes non voisines,
– (o), réactions qui ne sont pas présentes dans les autres espèces.

Le tableau 5.6 présente des réactions de l'espèce de référence ref et une autre espèce S.

Dans ce tableau, les cases en gras dans les colonnes R, "gènes de ref" et "gènes de S" désignent R' et les gènes voisins dans ref et S (voir le pied du tableau). R représente un ensemble de réactions de ref trouvées par les méthodes des chapitres précédents. Les symboles de la colonne S représentent des motifs métaboliques conservés entre les espèces ref et S pour les réactions de R. R' désigne un sous-ensemble maximal de R tel que les gènes de S impliqués dans les réactions de R' sont voisins.

R	**gènes de** ref	**gènes de** S	S
$\mathbf{r_2}$	**T**	$\mathbf{T_1}$	x
r_3	U	_	o
r_1	V	V_1	.
$\mathbf{r_9}$	**W**	$\mathbf{W_1}$	x
$\mathbf{R'}$	**gènes voisins**	**gènes voisins**	

TAB. 5.6 : La conservation par réactions pour l'espèce de référence ref et une autre espèce S (colonne S).

5.3.3 Conservation des trails par gènes

Deux gènes codant des enzymes impliquées dans la même réaction métabolique sont considérés comme fonctionnellement similaires. Des gènes fonctionnellement similaires dans deux espèces peuvent être soit des analogues (ont évolué séparément et ont une fonction similaire), soit des homologues (partageant une origine évolutive commune).

La conservation des *trails* par gènes consiste à construire le tableau dont les lignes représentent les gènes de l'espèce de référence ref impliquée dans réactions du support de chaque *trail* de ref partagés par ref et au moins une autre espèce dans P, et les colonnes représentent les autres espèces dans P (ordonnées en fonction de la distance évolutive par rapport à *ref*, comme dans la conservation des *trails* par réactions). Ce tableau reflète les motifs génomiques conservés entre l'espèce de référence et les autres espèces au travers deux symboles qui peuvent être attribués à chaque cellule. Ces symboles sont :

- (x), gènes voisines fonctionnellement similaires,
- (.), gènes non voisines fonctionnellement similaires.

Le tableau 5.7 présente des gènes voisins de l'espèce de référence ref et une autre espèce S. La conservation par gènes pour l'espèce de référence ref et une autre espèce S (colonne S). Les cases en gras dans les colonnes R, G et H désignent R' et les gènes voisins dans ref et S (voir le pied du tableau). R représente un ensemble de réactions de ref trouvées par les méthodes des chapitres précédents. Les symboles de la colonne S représentent des motifs génomiques conservés entre les espèces ref et S pour les réactions de R. G' désigne un sous-ensemble maximal de G tel que les gènes de S impliqués dans les réactions de R' sont voisins.

R	**G**	**H**	***S***
$\mathbf{r_2}$	**T**	$\mathbf{T_1}$	x
r_3	U	_	.
r_1	V	V_1	.
$\mathbf{r_9}$	**W**	$\mathbf{W_1}$	x
R'	**G'**	**H'**	

TAB. 5.7 : La conservation par gènes pour l'espèce de référence ref et une autre espèce S (colonne S).

Ces des méthodes ont été présentées dans [Zaharia et al., 2019].

5.3.4 Méthodes de résolution pour la conservation des trails

Nous avons adaptées les méthodes de résolution proposées dans le chapitre précédent pour résoudre le problème de la conservation des *trails* entre les espèces. Notre objectif est de remplir les tableaux présentés dans les sections précédentes. Le remplissage des tableaux se fait en plusieurs étapes. La première consiste à calculer les *trails* de l'espèce de référence ref. Plusieurs méthodes ont été développées pour calculer ces *trails*. La méthode la plus convenable est celle qui calcule les *trails* distincts. La première colonne du tableau est formée par les supports des *trails*. Une ligne vide dans le tableau séparant le support de *trail* i et $i+1$. Ensuite et à l'aide de nos méthodes proposées on en déduit la conservation des motifs métaboliques et génomiques entre l'espèce de référence et les autres espèces. Pour chaque support de *trail* calculé dans D et G de l'espèce de référence ref, on cherche dans D et G de $p \in P$ le *trail* qui maximise la couverture (maximum des "x"). Ces méthodes maximise les supports des *trails* de ref dans les autres espèces. Dans le cas de la conservation métabolique, la première colonne du tableau des résultats représente les supports des *trails* de l'espèce de référence et dans le cas de la conservation génomique, la première colonne du tableau des résultats représente l'ordre des gènes par brin et chromosome impliqués dans les supports des *trails*.

5.3.5 Analyse des résultats

Dans la section précédente, des méthodes ont été présentées pour maximiser les couvertures des supports des *trails* des autres espèces.

La présente section illustre l'analyse et l'interprétation de la conservation des motifs métaboliques et génomiques entre l'espèce de référence ref et les autres espèces.

Nous nous concentrons sur 50 espèces et nous désignons l'espèce *Escherichia coli* comme espèce de référence. Les supports de *trails* extraits dans l'espèce *Escherichia coli* peuvent être totalement conservés (trai commun), partiellement conservés (peut-être la résultat d'une évolution) ou non conservés (différence) dans les autres espèces.

La conservation des *trails* est effectuée pour l'espèce de référence, les autres espèces de l'ensemble de données sont classées par la distance évolutive croissant par rapport à l'espèce de référence. Si la distance évolutive entre deux espèces est faible (mesurée par le nombre de différences dans leurs séquences), alors elles susceptibles d'avoir un ancêtre commun récent ; mais si elles sont très éloignées, alors leur ancêtre commun se situe dans un passé lointain. La distance entre les espèces peut être utilisé comme une mesure de la distance dans le temps depuis que les espèces ont divergé.

Au fur et à mesure que le nombre d'espèces augmente, et que le nombre de séquences augmente, des données contradictoires commencent à apparaître. Le taux de mutation peut ne pas être suffisamment élevé pour distinguer les espèces étroitement apparentées.

La compréhension de l'évolution des espèces est un des enjeux clé de la biologie. Cela induit l'identification des événements génétiques affectant les génomes au cours du temps, mais aussi la détermination des relations entre les génomes, les phénotypes et l'environnement des organismes. A l'aide des résultats de conservation des *trails*, nous voulons fournir des éléments pour formuler des hypothèses, extraire de nouvelles connaissances et/ou de confirmer et/ou d'infirmer les connaissances issues de la phylogénie.

Nous pressentons ci-après quelques tableaux de conservations des *trails* en choisissant des espèces de distances évolutive faible et élevées.

Le tableau 5.8 montre la conservation d'un *trail* par réactions pour E. coli comme espèce de référence et 12 autres bactéries.

Dans ce tableau, les espèces *lac, snd, mpn* et *heo* ne possèdent pas toutes ou une grande majorité des réactions de ce *trail*. L' espèce *emi* à des gènes voisins fonctionnellement similaires voisins à deux gènes de E. coli impliqués dans trois réactions du *trail*. Les espèces *lmo, sau, mtv* et *sco* possèdent des gènes

voisins fonctionnellement similaires aux gènes de E. coli impliqués dans au moins quatre réactions de ce *trail*. Les espèces *ype, vco* et *spc* possèdent des gènes voisins fonctionnellement similaires aux gènes de E. coli impliqués dans toutes les réactions de ce *trail*. L'en-tête est formé par des groupes de classes bactériennes, à savoirs les classes *Alphaproteobacteria, Betaproteobacteria, Gammaproteobacteria, et Deltaproteobacteria, Terrabacteria, Sphingobacteria (bactéries FCB* et itPlanctobacteria (bactéries PVC).

reaction	eco_gene	ype	vco	spc	lmo	sau	lac	snd	mpn	mtv	sco	emi	heo
R03243	b2021	x	x	x	.	x	o	o	o	x	x	.	o
R03457	b2022	x	x	x	x	x	o	o	o	x	x	.	o
R01071	b2019	x	x	x	.	.	o	o	o	.	.	.	o
R04035	b2026	x	x	x	x	x	o	o	.	.	.	.	o
R03012	b2020	x	x	x	.	.	o	o	o	.	.	x	o
R03013	b2022	x	x	x	.	o	o	o	.	.	.	x	o
R04037	b2026	x	x	x	x	x	o	.	o	x	.	.	.
R01163	b2020	x	x	x	.	.	o	o	o	.	.	x	o
R04558	b2023	x	x	x	x	x	o	o	o	x	x	.	o
R04640	b2024	x	x	x	x	x	o	o	o	x	x	.	o

TAB. 5.8 : *Trail* de réactions *{R03243, R03457, R01071, R04035, R03012, R03013, R04037, R01163, R04558, R04640 }* catalysés parles gènes voisines *{b2021, b2022, b2019, b2026, b2020, b2022, b2026, b2020, b2023 , b2024 }*.

Le tableau 5.8 montre la conservation d'un *trail* par réactions pour E. coli comme espèce de référence et 12 autres bactéries. Ce *trail* est présent au total dans trois espèces en utilisant les produits de gènes voisins fonctionnellement similaires aux gènes de E. coli impliqués dans ce *trail*. En revanche, 5 espèces de l'ensemble de données (près de 33%) n'ont pas de gènes voisins fonctionnellement similaires aux gènes de E. coli impliqués dans ce *trail*. Le tableau complet est présenté dans l'Annexe A.

Le tableau 5.9 montre la conservation d'un *trail* par gènes pour E. coli comme espèce de référence et 12 autres bactéries.

Dans ce tableau, les espèces *lac, mtv, sco, emi* et *heo* ne possèdent pas des gènes fonctionnellement similaires impliqués aux gènes de E. Coli impliqués dans ce *trail*. L' espèce *mpn* a deux gènes voisins fonctionnellement similaires à deux gènes de E. coli impliqués dans dans ce *trail*. Les espèces *ype, vco, spc, lmo, sau* et *snd* possèdent des gènes voisins fonctionnellement similaires à quatre gènes de E. coli impliqués dans ce *trail*. Comme dans le cas de conservation des *trails* par reactions, l'en-tête est formée par des groupes de classes bactériennes, à savoirs les classes *Alphaproteobacteria, Betaproteobacteria, Gammaproteobacteria, et Deltaproteobacteria, Terrabacteria, Sphingobacteria (bactéries FCB* et itPlanctobacteria (bactéries PVC).

Le tableau 5.9 montre la conservation d'un *trail* par gènes pour E. coli comme espèce de référence et 12 autres bactéries. Le tableau complet est présenté dans l'Annexe A.

Le tableau 5.10 montre une comparaison entre les méthodes de conservations des *trails*. Notre méthode a trouvé plus de croix ne veut pas dire qu'on avait plus de *trails*; nous pouvons juste dire que notre famille de *trails* est de meilleure qualité.

De ces tableaux de conservations des motifs métaboliques et/ou génomiques, nous pouvons extraire de nouvelles connaissances et/ou de confirmer et/ou d'infirmer les connaissances issues de la phylogénie.

Contrairement à la méthode HNet, la méthode de calcul du tableau de conservation proposée par Zaharia *et al* dans [Zaharia et al., 2019] n'assure pas que les croix correspondent bien à des *trails* DG-

eco_gene	ype	vco	spc	lmo	sau	lac	snd	mpn	mtv	sco	emi	heo
b0115	x	x	x	x	x	o	x	.	o	o	o	o
b0118	.	.	.	.	.	o	o	o	o	o	o	o
b0114	x	x	x	x	x	o	x	x	o	o	o	o
b0116	x	x	x	x	x	o	x	.	o	o	.	o
b0118	.	.	.	.	.	o	o	o	.	.	o	.
b0114	x	x	x	x	x	o	x	x	o	o	o	o

TAB. 5.9 : *Trail* de réactions *{R02569, R01325, R00014, R07618, R01900, R03270 }* catalysés parles gènes voisines *{b0115, b0118 , b0114, b0116, b0118, b0114 }*

	Conservation des motifs [Zaharia et al., 2019]						Conservation des motifs (PPC)		
Famille de *trails*	Résultats de HNet [Zaharia et al., 2019]			Résultats de ppc			Résultats de ppc		
	x	.	o	x	.	o	x	.	o
eco (espèce de reference)	5216	6376	11879	5547	6431	12571	4969	6472	13108

TAB. 5.10 : Comparaison entre les méthodes de conservations des *trails*.

consistants dans les autres espèces ; il est donc normal que nous trouvions moins de croix ; nous pouvons en revanche espérer qu'elles soient porteuses de plus d'information.

5.4 Conclusion

Dans ce chapitre, nous avons développées des méthodes pour la recherche d'une famille de *trails*. Après avoir évaluer ces méthodes sur des données réelles nous avons montré comment les résultats propres à chaque espèce peuvent être exploités afin de détecter la conservation dans des espèces différentes.

Toutes les résultats des méthodes présentées dans les chapitres précédents peuvent être exploiter pour étudier la conservation des *trails*. Pour des raison biologique, nous avons choisi les résultats des méthodes qui permettent trouver une famille de *trails*. Nous avons présentés les méthodes développées pour chercher une famille de *trails* puis nous avons introduit le concept de conservation des *trails* présentés dans les travaux de Zaharia $et\ al.$ [Zaharia et al., 2019] par l'étude de conservation de deux points de vue. Premièrement la conservation des *trails* par réactions qui se concentre sur l'aspect métaboliques. En second, la conservation des trails par gènes qui se concentre sur l'aspect génomique. Enfin, nous avons présentés nos méthodes adaptés pour résoudre ce problème.

Chapitre 6

Conclusion et perspectives

Cette thèse s'intéresse à la mise en place d'outils pour la comparaison de graphes hétérogènes, avec des applications en bio-informatique. Elle aborde des problèmes liés aux réseaux biologiques hétérogènes. Elle se concentre sur la mise en évidence de relations entre métabolisme et contexte génomique par une approche d'exploration de graphes.

L'exploration de la relation entre le métabolisme et le contexte génomique répond aux deux objectifs principaux de cette thèse : la détection des motifs métaboliques et génomiques pour une seule espèce, d'une part, et l'étude de motifs métaboliques et génomiques conservés entre plusieurs espèces, d'autre part.

Nous avons choisi de nous concentrer sur la détection de réactions voisines catalysées par des produits de gènes voisins, où la notion de voisinage peut être modulée en permettant l'omission de plusieurs réactions et/ou gènes. Plus précisément, les motifs recherchés sont des *trails* de réactions (c'est-à-dire des chaines de réactions dans lesquelles les réactions peuvent être répétées) qui sont catalysées par des produits de gènes voisins. Le choix d'extraire les *trails* (plutôt que des chemins, notamment) a été motivé premièrement par l'importance (justifiée par Zaharia *et al* dans [Zaharia et al., 2019]) de n'ignorer ni l'orientation des réactions, ni la présence de circuits dans les réseaux métaboliques.

En plus de l'identification de motifs métaboliques et génomiques, nous étudions également le degré de conservation de ces motifs à l'échelle de plusieurs espèces. Une définition flexible de la conservation est adoptée. Ainsi, lors de l'évaluation de la conservation, l'ordre des réactions dans les *trails* et l'ordre des gènes fonctionnellement similaires sur le chromosome peuvent différer entre les espèces. De plus, la conservation peut être partielle, ce qui signifie que la composition des *trails* et des contextes génomiques peut varier, certaines espèces n'ayant conservé qu'une partie d'un motif métabolique et génomique détecté chez d'autres organismes.

Nos contributions ont consisté à proposer des méthodes et leurs implémentations pour résoudre des problèmes algorithmiquement difficiles, motivés par des problématiques biologiques.

Contributions

Après avoir brièvement présenté quelques notions de biologie dans le Chapitre 2, nous avons fait une synthèse de la littérature qui portent sur la comparaison des séquences et des réseaux pour la biologie. Nous avons ensuite présenté nos contributions dans les Chapitres 3, 4 et 5 .

À fin de détecter des motifs métaboliques et génomiques pour une espèce donnée, nous avons d'abord proposé dans le Chapitre 3 des méthodes exactes pour la résolution d'un problème de la littérature nommé One-To-One SkewGraM. Ce problème est issu d'un modèle de comparaison de réseaux hétérogènes identifié dans [Fertin et al., 2012, Babou, 2012]. Nous avons présenté pour ce modèle de comparaison, la formulation du problème, les méthodes de résolution proposées, et des expérimen-

tations sur des données aléatoires et des données réelles. Nous avons commencé par AlgoBB, un algorithme de *branch-and-bound* qui prend comme arguments un graphe orienté acyclique $D = (V, A)$ (un réseau métabolique), un graphe non-orienté $G = (V, E)$ (un graphe codant la proximité des gènes), et un arc (i, j) dans D. AlgoBB est composé de deux parties, dont la première, appelée $CoverSet$, calcule l'ensemble couvrant de (i, j), c'est-à-dire le sous-ensemble maximal de sommets qui pourraient étendre l'arc (i, j) à un chemin P de D tel que les sommets de P induisent un sous-graphe connexe dans G (c'est-à-dire un chemin satisfaisant la contrainte de proximité entre les gènes) passant par l'arc (i, j) [Babou, 2012]. Pour résoudre le problème posé, AlgoBB doit être appelé m fois, c'est-à-dire une fois par arc de D. Pour améliorer cet algorithme, nous avons proposé de nouvelles règles de dominance et une nouvelle politique de sélection pour les arcs qui nous permettent de résoudre le problème général plus efficacement. Ces améliorations permettent de traiter des des instances plus grandes que les résultats présentés dans [Fertin et al., 2012, Babou, 2012]. Nous avons également proposé deux formulations de programmation linéaire en nombres entiers (ILP) en adaptant et en combinant des formulations pour le problème du plus long chemin et le problème de l'arbre couvrant. Nous avons présenté ensuite un modèle de programmation par contraintes qui s'avère plus efficace que les autres méthodes. Ce travail a donné lieu à une publication dans une revue internationale [Ahmed Sidi et al., 2022] et des communications en conférences [Ahmed Sidi et al., 2019, Ahmed Sidi et al., 2020, Ahmed Sidi et al., 2021].

Dans le Chapitre 4, nous avons proposé des méthodes exactes pour la résolution d'un cas plus général du problème One-To-One SkewGraM. Nous avons étudié le cas où le graphe modélisant le réseau métabolique peut contenir des circuits. Nous avons proposé des méthodes exactes pour calculer un *trail* "maximum" de réactions dans un réseau métabolique, de sorte que ces réactions soient toutes catalysées par des produits de gènes voisins. Autrement dit, le *trail* de couverture maximale T dans un graphe orienté $D = (V, A)$, tel que les sommets du *trail* T induisent un sous-graphe connexe d'un graphe non-orienté $G = (V, E)$. Ces méthodes permettent de traiter des instances plus grandes que les résultats présentés dans [Zaharia et al., 2019].

Enfin, nous avons montré dans le Chapitre 5, comment les motifs métaboliques et génomiques propres à chaque espèce peuvent être exploités À fin d'évaluer leur degré de conservation à l'échelle de plusieurs espèces. Nous avons fait des rappels sur les concepts de conservation des *trails* présentés dans les travaux de Zaharia $et\ al.$ [Zaharia et al., 2019] (la conservation du point de vue métabolique ou génomique). Nous avons ensuite vu comment nos méthodes pouvaient être adaptées pour répondre à cette problématique particulière.

Perspectives

Les travaux de cette thèse nous ouvrent de nombreuses perspectives :

1. Explorer plus en avant ce problème de comparaison de graphes hétérogènes, en modélisant d'autres problèmes similaires, et en adaptant les différentes méthodes exactes développées.
2. Modéliser les critères de comparaison d'espèces dans les problèmes de recherche d'un plus long chemin et d'un plus long *trail*, en vue de trouver une famille de chemins ou de *trails* plus adaptée à cette comparaison.
3. Passer de la comparaison de deux réseaux hétérogènes à la comparaison de plus de deux réseaux hétérogènes. Une généralisation possible du problème SkewGraM est le problème de maximisation Multiple SkewGraM défini comme suit :

Multiple SkewGraM

Instance : Un ensemble de graphes orientés $\{D_1, D_2, ..., D_m\}$, un ensemble de graphes non-orientés $\{G_1, G_2, ..., G_m\}$ ayant tous le même ensemble de sommets $\{1, 2, ..., n\}$.

Solution : Un *trail* T de tous D_i, $1 < i < m$, tel que $G_i[V(T)]$, $1 < i < m$ est connexe.

Mesure : La couverture du *trail* T.

En raison des motivations algorithmiques et biologiques, il serait aussi possible de commencer par étudier un cas particulier du problème Multiple SkewGraM, que l'on pourrait appeler Multiple One-To-One SkewGraM, où D_i sont acycliques (DAG). Ce problème est formulé comme suit.

Multiple One-To-One SkewGraM

Instance : Un ensemble de DAGs $\{D_1, D_2, ..., D_m\}$, un ensemble de graphes non-orientés $\{G_1, G_2, ..., G_m\}$ ayant tous le même ensemble de sommets $\{1, 2, ..., n\}$.

Solution : Un chemin P de tous D_i, $1 < i < m$, tel que $G_i[V(P)]$, $1 < i < m$ est connexe.

Mesure : La longueur du chemin P.

4. Développer une plateforme de comparaison de réseaux biologiques permettant de faciliter aux biologistes l'utilisation des différentes méthodes développées. Du point de vue de l'utilisateur, il serait pratique d'avoir un outil de visualisation intégré qui met en évidence les résultats obtenus dans la recherche des motifs ou l'étude du dégré de conservation de ces motifs.
5. Développer des approches de résolution par décomposition car il est fort probable que les méthodes par décomposition soient efficaces pour les modèles linéaires. Les modèles linéaires proposés sont presque déjà (naturellement) séparables en deux sous-problèmes polynomiaux : recherche d'un chemin, recherche d'un arbre couvrant.
6. Mener des expérimentations sur des réseaux métaboliques complets.
7. Travailler sur les hypergraphes modélisant les réseaux métaboliques. La modélisation d'un réseau métabolique par un hypergraphe est une façon de résoudre l'ambiguïté des graphes simples est d'utiliser un hypergraphe. Pour rappel, un hypergraphe est un graphe dont les arêtes peuvent relier plus de deux noeuds. L'utilisation d'un hypergraphe permettrait, ainsi, de préciser quel(s) produit(s) exactement d'une (d'un ensemble de) réaction(s) est (sont) le(s) substrat(s) d'une réaction donnée.
8. Ignorer l'orientation dans D comme dans [Ogata et al., 2000] et chercher un sous-graphe connexe dans D tel que les sommets de ce sous-graphe induisent un sous-graphe connexe dans G (c'est-à-dire un sous-graphe satisfaisant la contrainte de proximité entre les gènes).
9. Traiter des problèmes similaires tels que la construction des voies métaboliques. Par exemple, pour combler le manque de connaissances des organismes peu étudiés (c'est-à-dire l'incomplétude des données de leur réseau métabolique, de leur réseau d'interactions protéine-protéine, ...), des méthodes ont été développées pour suggérer des réactions permettant de compléter les réseaux métaboliques d'espèces moins étudiées grâce à des réactions apparaissant dans des réseaux d'espèces apparentées bien connues. Il existe de nombreux problèmes de complétion de réseaux [Prigent et al., 2017].

Au cours de mes derniers mois de thèse, j'ai eu l'occasion d'encadrer un stage de Master en biologie portant sur l'analyse et l'interprétation des familles de *trails* que nous générons. Les résultats obtenus à cette occasion doivent encore être consolidés. En outre, nous avons récemment noué des liens avec

des biologistes travaillant sur la complétion de réseau métabolique. Nous pensons donc, dans un futur proche, nous concentrer sur le point 9.

Annexe A

Conservations des motifs métaboliques et génomiques

Le tableau A.1 montre la conservation d'un *trail* par réactions pour E. coli comme espèce de référence.

reaction	eco_gene	ype	vco	spc	pae	xfa	rso	nme	afi	ara	rrj	gsu	nde	aca	din	fnu	dap	tid	aae	bsu	lmo	sau	lac	snd	cpe	mpn	syn	pma	cau	bbv	cgl	mtv	sco	dra	tth	fgi	amo	tmm	cex	dth	fsu	gau	cph	bfr	rba	cpn	ote	bbn	emi	heo
R03243	b2021	x	x	x	.	x	x	.	x	.	o	x	.	o	.	o	.	.	o	.	.	x	o	o	o	o	.	.	x	x	x	x	x	.	x	.	.	x	o	x	x	.	.	x	.	.	x	o	.	o
R03457	b2022	x	x	x	x	x	x	.	x	x	o	x	x	o	x	o	x	.	o	x	x	x	o	o	o	o	.	.	x	x	x	x	x	x	x	x	x	x	o	x	.	x	.	x	.	o	x	o	x	o
R01071	b2019	x	x	x	.	x	.	.	x	.	o	x	x	o	.	o	.	.	o	x	x	x	o	o	o	o	.	.	.	.	.	.	.	.	.	.	x	x	o	x	.	x	.	.	x	o	.	o	x	o
R04035	b2026	x	x	x	.	x	x	.	x	x	o	x	x	o	.	o	x	o	o	x	x	x	o	o	o	.	.	.	.	.	.	.	x	x	o	x	x	o	x	.	x	o	.	x	.	o	o	x	o
R03012	b2020	x	x	x	.	x	.	.	x	.	o	.	.	o	.	o	.	.	o	x	x	.	o	o	o	o	.	.	.	.	x	x	x	.	.	.	.	x	o	x	.	.	.	x	.	o	.	o	x	o
R03013	b2022	x	x	x	o	x	o	o	.	.	o	o	o	.	.	.	o	o	o	.	.	o	o	o	.	.	o	o	o	o	.	.	.	.	.	o	o	.	.	.	o	o	o	x	o	.	o	o	x	o
R04037	b2026	x	x	x	.	x	x	x	x	.	o	x	x	o	.	o	.	.	o	x	x	x	o	o	o	o	.	.	.	x	x	x	x	x	x	.	x	x	o	x	.	x	.	.	.	o	.	o	x	o
R01163	b2020	x	x	x	.	x	.	.	x	.	o	.	.	o	.	o	.	.	o	x	x	.	o	o	o	o	.	.	.	.	x	x	x	.	.	.	.	x	o	x	.	.	.	x	.	o	.	o	x	o
R04558	b2023 b2025	x	x	x	x	x	x	x	x	x	o	x	x	o	x	o	x	.	o	x	x	x	o	o	o	o	.	.	.	x	x	x	x	x	x	x	x	x	o	x	x	x	x	.	.	o	x	o	.	o
R04640	b2024	x	x	x	x	x	x	x	x	x	o	x	x	o	x	o	x	.	o	x	x	x	o	o	o	o	.	.	.	x	x	x	x	.	.	x	x	x	o	x	.	x	x	.	.	o	.	o	x	o

TAB. A.1 : *Trail* de réactions *{R03243, R03457, R01071, R04035, R03012, R03013, R04037, R01163, R04558, R04640 }* catalysés parles gènes voisines *{b2021, b2022, b2019, b2026, b2020, b2022, b2026, b2020, b2023 , b2024 }*.

Le tableau A.2 montre la conservation d'un *trail* par gènes pour E. coli comme espèce de référence.

reaction	eco_gene	ype	vco	spc	pae	xfa	rso	nme	afi	ara	rrj	gsu	nde	aca	din	fnu	dap	tid	aae	bsu	lmo	sau	lac	snd	cpe	mpn	syn	pma	cau	bbv	cgl	mtv	sco	dra	tth	fgi	amo	tmm	cex	dth	fsu	gau	cph	bfr	rba	cpn	ote	bbn	emi	heo
R02569	b0115	x	x	x	x	x	x	x	x	x	.	x	x	o	o	o	o	o	o	x	x	x	o	x	o	x	.	.	x	o	o	o	o	x	x	.	.	o	.	.	o	x	o	o	x	x	x	o	o	o
R01325	b0118 b0771 b1276	x	.	x	.	.	.	.	.	.	.	x	.	.	.	o	.	.	.	.	.	.	.	o	o	o	o	.	.	.	.	o	o	o	.	.	.	o	o	o	o	.	.	.	.	o	.	o	o	o
R00014	b0114	x	x	x	x	.	x	x	x	x	x	x	x	.	o	o	o	o	o	x	x	x	o	x	o	x	.	.	x	o	o	o	o	x	.	x	o	o	o	o	o	x	o	o	x	x	x	o	o	o
R07618	b0116	x	x	x	.	x	x	x	x	x	.	x	.	.	.	o	.	o	.	x	x	x	o	x	o	x	.	.	.	.	o	o	o	.	x	.	.	.	.	.	o	x	.	.	.	.	.	o	.	o
R01900	b0118 b0771 b1276	x	.	x	.	.	.	.	.	.	.	x	.	.	.	o	.	.	.	.	.	.	.	o	o	o	o	.	.	.	.	.	.	.	.	.	.	o	o	o	o	.	.	.	.	o	.	o	o	.
R03270	b0114	x	x	x	x	.	x	x	x	x	x	x	x	.	o	o	o	o	o	x	x	x	o	x	o	x	.	.	x	o	o	o	o	x	.	x	o	o	o	o	o	x	o	o	x	x	x	o	o	o

TAB. A.2 : *Trail* de réactions *{R02569, R01325, R00014, R07618, R01900, R03270}* catalysés parles gènes voisines *{b0115, b0118 , b0114, b0116, b0118, b0114}*

Bibliographie

[Abdelmaguid, 2018] Abdelmaguid, T. F. (2018). An efficient mixed integer linear programming model for the Minimum spanning tree problem. *Mathematics*, 6(10) :1–17.

[Aderem, 2005] Aderem, A. (2005). Systems biology : Its practice and challenges. *Cell*, 121(4) :511–513.

[Ahmed Sidi et al., 2019] Ahmed Sidi, M. L., Bocquillion, R., Babou, H. M., Dhib, C., Nanne, M. F., Neron, E., and Soukhal, A. (2019). Méthodes exactes pour la détermination d'un plus long chemin dg-consistant dans des réseaux biologiques. In *ROADEF 2019, 20ème congrès annuel de la société Française de Recherche Opérationnelle et dAide à la Décision*.

[Ahmed Sidi et al., 2021] Ahmed Sidi, M. L., Bocquillon, R., Babou, H., Dhib, C., Néron, E., Soukhal, A., and Nanne, M. (2021). Un modèle de programmation par contraintes pour la recherche d'un chemin dg-consistant dans des réseaux biologiques. In *22ème conférence de la société Française de Recherche Opérationnelle et dAide à la Décision (ROADEF)*.

[Ahmed Sidi et al., 2020] Ahmed Sidi, M. L., Bocquillon, R., Babou, H. M., Dhib, C., Nanne, M. F., Neron, E., and Soukhal, A. (2020). Exact methods for finding a longest dg-consistent trail in biological networks. Technical report, EasyChair.

[Ahmed Sidi et al., 2022] Ahmed Sidi, M. L., Bocquillon, R., Mohamed Babou, H., Dhib, C., Néron, E., Soukhal, A., and Nanne, M. F. (2022). Improved approaches to solve the one-to-one skewgram problem. *Computers & Operations Research*, 138 :105584.

[Akihiro Nakaya and Kanehisa, 2001] Akihiro Nakaya, S. G. and Kanehisa, M. (2001). Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis. *Genome Informatics*, 53 :44–53.

[Albert and Barabási, 2002] Albert, R. and Barabási, A. L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1) :47–97.

[Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3) :403–410.

[Alves et al., 2002] Alves, R., Chaleil, R. A., and Sternberg, M. J. (2002). Evolution of enzymes in metabolism : A network perspective. *Journal of Molecular Biology*, 320(4) :751–770.

[Apic et al., 2001] Apic, G., Gough, J., and Teichmann, S. A. (2001). Domain combinations in archaeal, eubacterial and eukaryotic proteomes. *Journal of Molecular Biology*, 310(2) :311–325.

[Arita, 2004] Arita, M. (2004). The metabolic world of Escherichia coli is not small. *Proceedings of the National Academy of Sciences of the United States of America*, 101(6) :1543–1547.

[Babou, 2012] Babou, H. M. (2012). *Comparaison de réseaux biologiques biologiques*.

[Baewicz et al., 2005] Baewicz, J., Formanowicz, P., and Kasprzak, M. (2005). Selected combinatorial problems of computational biology. *European Journal of Operational Research*, 161(3) :585–597.

[Barabã et al., 2002] Barabã, A. L., Barabâsi, A.-L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., and Vicsek, T. (2002). Evolution of the social network of scientific collaborations. *Physica A : Statistical mechanics and its applications*, 311(3) :590–614.

[Barabasi, 2009] Barabasi, A.-L. (2009). Scale-free networks : A decade and beyond. *Science*, 325(5939) :412–413.

[Barabasi and Albert, 1999] Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439) :509–512.

[Barabasi and Oltvai, 2004] Barabasi, A.-L. and Oltvai, Z. N. (2004). Network biology : understanding the cell's functional organization. *Nature reviews genetics*, 5(2) :101–113.

[Barrat and Weigt, 2001] Barrat, A. and Weigt, M. (2001). On the properties of small-world network models. *European Physical Journal B*, 22(1) :3–10.

[Barratt et al., 1990] Barratt, C., Bolton, A., and Cooke, I. (1990). Functional significance of white blood cells in the male and female reproductive tract. *Human Reproduction*, 5(6) :639–648.

[Bernfeld et al., 1948] Bernfeld, P., Staub, A., and Fischer, E. H. (1948). Sur les enzymes amylolytiques xi. propriétés de l'-amylase de salive humaine cristallisée. *Helvetica Chimica Acta*, 31(7) :2165–2172.

[Blazewicz et al., 2018] Blazewicz, J., Kasprzak, M., Kierzynka, M., Frohmberg, W., Swiercz, A., Wojciechowski, P., and Zurkowski, P. (2018). Graph algorithms for dna sequencing origins, current models and the future. *European Journal of Operational Research*, 264(3) :799–812.

[Boccaletti et al., 2006] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D. U. (2006). Complex networks : Structure and dynamics. *Physics Reports*, 424(4-5) :175–308.

[Bordron et al., 2011] Bordron, P., Eveillard, D., and Rusu, I. (2011). SIPPER : A flexible method to integrate heterogeneous data into a metabolic network. *2011 IEEE 1st International Conference on Computational Advances in Bio and Medical Sciences, ICCABS 2011*, (June) :40–45.

[Bouamama et al., 2019] Bouamama, S., Blum, C., and Fages, J.-G. (2019). An algorithm based on ant colony optimization for the minimum connected dominating set problem. *Applied Soft Computing*, 80 :672–686.

[Boyer et al., 2005] Boyer, F., Morgat, A., Labarre, L., Pothier, J., and Viari, A. (2005). Syntons, metabolons and interactons : An exact graph-theoretical approach for exploring neighbourhood between genomic and functional data. *Bioinformatics*, 21(23) :4209–4215.

[Bramwell et al., 1988] Bramwell, V. H., Santoro, A., Rouesse, J., Mouridsen, H., Steward, W., Van Oosterom, A., Somers, R., Buesa, J., Mulder, J., Schutte, J., Pinedo, H., Blackledge, G., Wagener, T., and Dombernowky, P. (1988). Review of the clinical trials activity of the soft tissue and bone sarcoma group of the european organization for research and treatment of cancer. *Seminars in Surgical Oncology*, 4(1) :45–52.

[Bruggeman and Westerhoff, 2007] Bruggeman, F. and Westerhoff, H. (2007). The nature of systems biology. *Trends in microbiology*, 15 1 :45–50.

[Bunke, 2000] Bunke, H. (2000). Graph matching : Theoretical foundations, algorithms, and applications. In *In Proceedings of Vision Interface 2000, Montreal*, pages 82–88.

[Captivo et al., 2009] Captivo, M., Clímaco, J. C., and Pascoal, M. M. (2009). A mixed integer linear formulation for the minimum label spanning tree problem. *Computers &Operations Research*, 36(11) :3082–3085.

[Carlsson, 2009] Carlsson, M. (2009). SICStus Prolog User ' s Manual. *Reading*, (December).

[Chahira, 2013] Chahira, D. (2013). Table des matières Table des matières. *Réforme, Humanisme, Renaissance*, 15(2) :83.

[Chan and Loscalzo, 2012] Chan, S. Y. and Loscalzo, J. (2012). The emerging paradigm of network medicine in the study of human disease. *Circulation research*, 111(3) :359–374.

[Christakos et al., 2010] Christakos, S., Ajibade, D. V., Dhawan, P., Fechner, A. J., and Mady, L. J. (2010). Vitamin D : Metabolism. *Endocrinology and Metabolism Clinics of North America*, 39(2) :243–253.

[Ciriello and Guerra, 2008] Ciriello, G. and Guerra, C. (2008). A review on models and algorithms for motif discovery in protein-protein interaction networks. *Briefings in Functional Genomics and Proteomics*, 7(2) :147–156.

[Cohen et al., 2001] Cohen, R., Erez, K., Ben-Avraham, D., and Havlin, S. (2001). Breakdown of the internet under intentional attack. *Physical review letters*, 86(16) :3682.

[Collins et al., 1998] Collins, F. S., Patrinos, A., Jordan, E., Chakravarti, A., Gesteland, R., and Walters, L. R. (1998). New goals for the U.S. Human Genome Project : 1998-2003. *Science*, 282(5389) :682–689.

[Conte et al., 2004] Conte, D., FOGGIA, P., SANSONE, C., and VENTO, M. (2004). Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03) :265–298.

[Cottret et al., 2018] Cottret, L., Frainay, C., Chazalviel, M., Cabanettes, F., Gloaguen, Y., Camenen, E., Merlet, B., Heux, S., Portais, J.-C., Poupin, N., Vinson, F., and Jourdan, F. (2018). MetExplore : collaborative edition and exploration of metabolic networks. *Nucleic Acids Research*, 46(W1) :W495–W502.

[Dawes and Ribbons, 1964] Dawes, E. A. and Ribbons, D. W. (1964). Some Aspects of the Endogenous Metabolism of Bacteria. *Bacteriological reviews*, 28(2) :126–149.

[Dechter, 1990] Dechter, R. (1990). Enhancement schemes for constraint processing : Backjumping, learning, and cutset decomposition. *Artificial Intelligence*, 41(3) :273–312.

[Dechter and Meiri, 1989] Dechter, R. and Meiri, I. (1989). *Experimental evaluation of preprocessing techniques in constraint satisfaction problems.* Citeseer.

[Dias et al., 2017] Dias, F. C., Campêlo, M., Souza, C., and Andrade, R. (2017). Min-degree constrained minimum spanning tree problem with fixed centrals and terminals : Complexity, properties and formulations. *Computers &Operations Research*, 84 :46–61.

[Durek and Walther, 2008] Durek, P. and Walther, D. (2008). The integrated analysis of metabolic and protein interaction networks reveals novel molecular organizing principles. *BMC Systems Biology*, 2 :1–20.

[Elmsallati et al., 2018] Elmsallati, A., Msalati, A., and Kalita, J. (2018). Index-Based Network Aligner of Protein-Protein Interaction Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(1) :330–336.

[Emmert-Streib et al., 2016] Emmert-Streib, F., Dehmer, M., and Shi, Y. (2016). Fifty years of graph matching, network alignment and network comparison. *Information Sciences*, 346-347 :180–197.

[Evans et al., 2011] Evans, G. E., Sofronov, G. Y., Keith, J. M., and Kroese, D. P. (2011). Estimating change-points in biological sequences via the cross-entropy method. *Annals of Operations Research*, 189(1) :155–165.

[Fages, 2014] Fages, J.-G. (2014). Exploitation de structures de graphe en programmation par contraintes.

[Fages et al., 2016] Fages, J.-G., Lorca, X., and Rousseau, L.-M. (2016). The salesman and the tree : the importance of search in cp. *Constraints*, 21(2) :145–162.

[Fertin et al., 2012] Fertin, G., Mohamed Babou, H., and Rusu, I. (2012). Algorithms for subnetwork mining in heterogeneous networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7276 LNCS :184–194.

[Fields and Song, 1989] Fields, S. and Song, O. K. (1989). A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230) :245–246.

[Frolkis et al., 2009] Frolkis, A., Knox, C., Lim, E., Jewison, T., Law, V., Hau, D. D., Liu, P., Gautam, B., Ly, S., Guo, A. C., Xia, J., Liang, Y., Shrivastava, S., and Wishart, D. S. (2009). SMPDB : The Small Molecule Pathway Database. *Nucleic Acids Research*, 38 :D480–D487.

[Girvan and Newman, 2002] Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12) :7821–7826.

[Gusfield, 1997] Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences : Computer Science and Computational Biology*. Cambridge University Press.

[Guzzi and Milenković, 2018] Guzzi, P. H. and Milenković, T. (2018). Survey of local and global biological network alignment : The need to reconcile the two sides of the same coin. *Briefings in Bioinformatics*, 19(3) :472–481.

[Horowitz, 1945] Horowitz, N. H. (1945). On the Evolution of Biochemical Syntheses. *Proceedings of the National Academy of Sciences*, 31(6) :153–157.

[Huthmacher et al., 2008] Huthmacher, C., Gille, C., and Holzhütter, H.-G. (2008). A computational analysis of protein interactions in metabolic networks reveals novel enzyme pairs potentially involved in metabolic channeling. *Journal of Theoretical Biology*, 252(3) :456–464. In Memory of Reinhart Heinrich.

[Ihmels et al., 2004] Ihmels, J., Levy, R., and Barkai, N. (2004). Principles of transcriptional control in the metabolic network of Saccharomyces cerevisiae. *Nature Biotechnology*, 22(1) :86–92.

[Istrail et al., 2012] Istrail, E. S., Pevzner, P., and Waterman, M. (2012). *Lecture Notes in Bioinformatics*.

[Ito et al., 2001] Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., and Sakaki, Y. (2001). A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the United States of America*, 98(8) :4569–4574.

[Jensen and Meckling, 1976] Jensen, M. C. and Meckling, W. H. (1976). Theory of the firm : Managerial behavior, agency costs and ownership structure. *Journal of Financial Economics*, 3(4) :305–360.

[Jeong and Albert, 2000] Jeong, H. and Albert, R. (2000). 35036627. 760(1990) :651–654.

[Jeong et al., 2001] Jeong, H., Mason, S. P., Barabási, A. L., and Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, 411(6833) :41–42.

[Junker, 2001] Junker, U. (2001). Quickxplain : Conflict detection for arbitrary constraint propagation algorithms. In *IJCAI01 Workshop on Modelling and Solving problems with constraints*, volume 4. Citeseer.

[Kanehisa et al., 2017] Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., and Morishima, K. (2017). KEGG : New perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research*, 45(D1) :D353–D361.

[Kanehisa and Goto, 2000] Kanehisa, M. and Goto, S. (2000). KEGG : Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1) :27–30.

[Kanehisa et al., 2013] Kanehisa, M., Goto, S., Sato, Y., Kawashima, M., Furumichi, M., and Tanabe, M. (2013). Data, information, knowledge and principle : back to metabolism in KEGG. *Nucleic Acids Research*, 42(D1) :D199–D205.

[Kelley et al., 2003] Kelley, B. P., Sharan, R., Karp, R. M., Sittler, T., Root, D. E., Stockwell, B. R., and Ideker, T. (2003). Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proceedings of the National Academy of Sciences*, 100(20) :11394–11399.

[Kelley et al., 2004] Kelley, B. P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B. R., and Ideker, T. (2004). PathBLAST : A tool for alignment of protein interaction networks. *Nucleic Acids Research*, 32(WEB SERVER ISS.) :83–88.

[Kern and Zuiderweg, 2003] Kern, D. and Zuiderweg, E. R. (2003). The role of dynamics in allosteric regulation. *Current Opinion in Structural Biology*, 13(6) :748–757.

[Kevans et al., 2015] Kevans, D., Turpin, W., Madsen, K., Meddings, J., Shestopaloff, K., Xu, W., Moreno-Hagelsieb, G., Griffiths, A., Silverberg, M. S., Paterson, A., et al. (2015). Determinants of intestinal permeability in healthy first-degree relatives of individuals with crohn's disease. *Inflammatory bowel diseases*, 21(4) :879–887.

[Kharchenko et al., 2004] Kharchenko, P., Vitkup, D., and Church, G. M. (2004). Filling gaps in a metabolic network using expression information. *Bioinformatics*, 20(SUPPL. 1) :178–185.

[Klau, 2009] Klau, G. W. (2009). A new graph-based method for pairwise global network alignment. *BMC Bioinformatics*, 10(SUPPL. 1).

[Koyutürk et al., 2004] Koyutürk, M., Grama, A., and Szpankowski, W. (2004). An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics*, 20(SUPPL. 1) :200–207.

[Kuchaiev et al., 2010] Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., and Pržulj, N. (2010). Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, 7(50) :1341–1354.

[Kuchaiev and Prulj, 2011] Kuchaiev, O. and Prulj, N. (2011). Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 27(10) :1390–1396.

[Laborie et al., 2018] Laborie, P., Rogerie, J., Shaw, P., and Vilím, P. (2018). IBM ILOG CP optimizer for scheduling : 20+ years of scheduling with constraints at IBM/ILOG. *Constraints*, 23(2) :210–250.

[Lacroix et al., 2008] Lacroix, V., Cottret, L., Thébault, P., and Sagot, M. F. (2008). An introduction to metabolic networks and their structural analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(4) :594–617.

[Lazcano and Miller, 1996] Lazcano, A. and Miller, S. L. (1996). The origin and early evolution of life : Prebiotic chemistry, the pre-RNA world, and time. *Cell*, 85(6) :793–798.

[Lee et al., 2004] Lee, I., Date, S. V., Adai, A. T., and Marcotte, E. M. (2004). R EPORTS Network of Yeast Genes. *Science*, 306(November) :1555–1558.

[Liao et al., 2009] Liao, C. S., Lu, K., Baym, M., Singh, R., and Berger, B. (2009). IsoRankN : Spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12) :253–258.

[Loch and Faisal, 2015] Loch, T. P. and Faisal, M. (2015). Emerging flavobacterial infections in fish : A review. *Journal of Advanced Research*, 6(3) :283–300.

[Loweth, 1997] Loweth, R. P. (1997). *Introduction to computations*.

[Ma and Liao, 2020] Ma, C.-Y. and Liao, C.-S. (2020). A review of proteinprotein interaction network alignment : From pathway comparison to global alignment. *Computational and Structural Biotechnology Journal*, 18 :2647–2656.

[Maguire et al., 2015] Maguire, R., Ream, E., Richardson, A., Connaghan, J., Johnston, B., Kotronoulas, G., Pedersen, V., McPhelim, J., Pattison, N., Smith, A., et al. (2015). Development of a novel remote patient monitoring system : the advanced symptom management system for radiotherapy to improve the symptom experience of patients with lung cancer receiving radiotherapy. *Cancer nursing*, 38(2) :E37–E47.

[Malapert, 2011] Malapert, A. (2011). *Techniques d'ordonnancement d'atelier et de fournées basées sur la programmation par contraintes*. PhD thesis, Université de Nantes ; Ecole Centrale de Nantes (ECN).

[Martin, 1991] Martin, R. (1991). Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters*, 10(3) :119–128.

[Matsuo et al., 2006] Matsuo, Y., Hamasaki, M., Nakamura, Y., Nishimura, T., Hasida, K., Takeda, H., Mori, J., Bollegala, D., and Ishizuka, M. (2006). Spinning multiple social networks for Semantic Web. *Proceedings of the National Conference on Artificial Intelligence*, 2 :1381–1387.

[Meng et al., 2016] Meng, S., Wu, M., Wang, Q., Dai, Z., Si, W., Huang, W., and Dong, X. (2016). Cobalt oxide nanosheets wrapped onto nickel foam for non-enzymatic detection of glucose. *Nanotechnology*, 27(34) :344001.

[Milo et al., 2004] Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M., and Alon, U. (2004). Superfamilies of Evolved and Designed Networks. *Science*, 303(5663) :1538–1542.

[Milo et al., 2002] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs : Simple building blocks of complex networks. *Science*, 298(5594) :824–827.

[Moreno-Hagelsieb and Santoyo, 2015] Moreno-Hagelsieb, G. and Santoyo, G. (2015). Predicting functional interactions among genes in prokaryotes by genomic context. *Prokaryotic Systems Biology*, pages 97–106.

[Mousavi and Tabataba, 2012] Mousavi, S. R. and Tabataba, F. (2012). An improved algorithm for the longest common subsequence problem. *Comput. Oper. Res.*, 39(3) :512520.

[Myers, 1991] Myers, E. W. (1991). An overview of sequence comparison algorithms in molecular biology. pages 1–25.

[Nobeli et al., 2009] Nobeli, I., Favia, A. D., and Thornton, J. M. (2009). Protein promiscuity and its implications for biotechnology. *Nature Biotechnology*, 27(2) :157–167.

[Ogata et al., 2000] Ogata, H., Fujibuchi, W., Goto, S., and Kanehisa, M. (2000). A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Research*, 28(20) :4021–4028.

[Ogata et al., 1999] Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG : Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 27(1) :29–34.

[Overbeek et al., 1999] Overbeek, R., Fonstein, M., Dsouza, M., Pusch, G. D., and Maltsev, N. (1999). The use of gene clusters to infer functional coupling. *Proceedings of the National Academy of Sciences*, 96(6) :2896–2901.

[Pearson and Lipman, 1988] Pearson, W. R. and Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the United States of America*, 85(8) :2444–2448.

[Pereira, 2018] Pereira, C. (2018). Nouvelles approches bioinformatiques pour l ' étude à grande échelle de l ' évolution des activités enzymatiques To cite this version : HAL Id : tel-01839673 Nouvelles approches bioinformatiques pour l ' étude à grande échelle de l ' évolution des activit.

[Pál and Hurst, 2004] Pál, C. and Hurst, L. D. (2004). Evidence against the selfish operon theory. *Trends in Genetics*, 20(6) :232–234. Pagination error in this issue, see Publisher's note in Vol. 21 issue 1 p. 36.

[Prigent et al., 2017] Prigent, S., Frioux, C., Dittami, S. M., Thiele, S., Larhlimi, A., Collet, G., Gutknecht, F., Got, J., Eveillard, D., Bourdon, J., et al. (2017). Meneco, a topology-based gap-filling tool applicable to degraded genome-wide metabolic networks. *PLoS computational biology*, 13(1) :e1005276.

[Prud'homme et al., 2017] Prud'homme, C., Fages, J.-G., and Lorca, X. (2017). Choco3 Documentation.

[Ravasz et al., 2002] Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A. L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586) :1551–1555.

[Renyi, 1959] Renyi, E. (1959). On random graph. *Publicationes Mathematicate*, 6 :290–297.

[Rison et al., 2002] Rison, S. C., Teichmann, S. A., and Thornton, J. M. (2002). Homology, pathway distance and chromosomal localization of the small molecule metabolism enzymes in escherichia coli. *Journal of Molecular Biology*, 318(3) :911–932.

[Rubinov and Sporns, 2010] Rubinov, M. and Sporns, O. (2010). Complex network measures of brain connectivity : Uses and interpretations. *NeuroImage*, 52(3) :1059–1069.

[Samal et al., 2006] Samal, A., Singh, S., Giri, V., Krishna, S., Raghuram, N., and Jain, S. (2006). Low degree metabolites explain essential reactions and enhance modularity in biological netwoks. *BMC Bioinformatics*, 7 :1–10.

[Sharan et al., 2005] Sharan, R., Suthram, S., Kelley, R. M., Kuhn, T., McCuine, S., Uetz, P., Sittler, T., Karp, R. M., and Ideker, T. (2005). Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences of the United States of America*, 102(6) :1974–1979.

[Sharan et al., 2007] Sharan, R., Ulitsky, I., and Shamir, R. (2007). Network-based prediction of protein function. *Molecular Systems Biology*, 3(88) :1–13.

[Shen-Orr et al., 2002] Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, 31(1) :64–68.

[Shyu and Tsai, 2009] Shyu, S. and Tsai, C.-Y. (2009). Finding the longest common subsequence for multiple biological sequences by ant colony optimization. *Computers & Operations Research*, 36 :73–91.

[Silvestri et al., 2017] Silvestri, S., Laporte, G., and Cerulli, R. (2017). A branch-and-cut algorithm for the minimum branch vertices spanning tree problem. *Computers &Operations Research*, 81 :322–332.

[Spirin et al., 2006] Spirin, V., Gelfand, M. S., Mironov, A. A., and Mirny, L. A. (2006). A metabolic network in the evolutionary context : Multiscale structure and modularity. *Proceedings of the National Academy of Sciences of the United States of America*, 103(23) :8774–8779.

[Steuer and Lopez, 2008] Steuer, R. and Lopez, G. Z. (2008). Global network properties. *Analysis of biological networks*, 31 :63.

[Strogatz, 2001] Strogatz, S. H. (2001). Exploring complex networks. *Nature*, 410(6825) :268–276.

[Tilk and Irnich, 2018] Tilk, C. and Irnich, S. (2018). Combined column-and-row-generation for the optimal communication spanning tree problem. *Computers &Operations Research*, 93 :113–122.

[Tipton and Boyce, 2000] Tipton, K. and Boyce, S. (2000). History of the enzyme nomenclature system. *Bioinformatics*, 16(1) :34–40.

[Towfic et al., 2009] Towfic, F., Greenlee, M. H. W., and Honavar, V. (2009). Aligning Biomolecular Networks Using Modular Graph Kernels. pages 345–361.

[Waterman, 1998] Waterman, M. S. (1998). Introduction to Computational Biology : Maps, Sequences and Genomes. *Biometrics*, 54(1) :398.

[Watts and Strogatz, 1998] Watts, D. J. and Strogatz, S. H. (1998). Strogatz - small world network Nature. *Nature*, 393(June) :440–442.

[Webb, 1992] Webb, C. (1992). The Use of the First Person in Academic Writing. *Journal of Advanced Nursing*, 17 :747–752.

[Wernicke and Rasche, 2007] Wernicke, S. and Rasche, F. (2007). Simple and fast alignment of metabolic pathways by exploiting local diversity. *Bioinformatics*, 23(15) :1978–1985.

[Wittig and De Beuckelaer, 2001] Wittig, U. and De Beuckelaer, A. (2001). Analysis and comparison of metabolic pathway databases. *Briefings in Bioinformatics*, 2(2) :126–142.

[Yang et al., 2021] Yang, X., Gao, S., Guo, L., Wang, B., Jia, Y., Zhou, J., Che, Y., Jia, P., Lin, J., Xu, T., Sun, J., and Ye, K. (2021). Three chromosome-scale Papaver genomes reveal punctuated patchwork evolution of the morphinan and noscapine biosynthesis pathway. *Nature communications*, 12(1) :6030.

[Zaharia, 2018] Zaharia, A. (2018). *Mining conserved neighborhood patterns in metabolic and genomic contexts*. PhD thesis, Université Paris Saclay (COmUE).

[Zaharia et al., 2019] Zaharia, A., Labedan, B., Froidevaux, C., and Denise, A. (2019). CoMetGeNe : Mining conserved neighborhood patterns in metabolic and genomic contexts. *BMC Bioinformatics*, 20(1) :1–20.

[Zakowski and Bruns, 1985] Zakowski, J. J. and Bruns, D. E. (1985). Biochemistry of human alpha amylase isoenzymes. *CRC Critical Reviews in Clinical Laboratory Sciences*, 21(4) :283–322.

[Zheng et al., 2003] Zheng, Y., Szustakowski, J. D., Fortnow, L., Roberts, R. J., and Kasif, S. (2003). Computational identification of operons in microbial genomes. *Genome Research*, 13(1) :1221–1230.