Analyse des systèmes bactériens: une approche in silico pour intégrer les connaissances du vivant

Philippe Bordron

UNIVERSITÉ DE NANTES

UFR DES SCIENCES ET TECHNIQUES

SCIENCES ET TECHNOLOGIES
DE L'INFORMATION ET DE MATHÉMATIQUES

Année 2012

N° attribué par la bibliothèque

Analyse des systèmes bactériens

une approche in silico *pour intégrer les connaissances du vivant*

THÈSE DE DOCTORAT
Discipline : INFORMATIQUE
Spécialité : INFORMATIQUE

Présentée
et soutenue publiquement par

Philippe BORDRON

le 27 mars 2012 au LINA, devant le jury ci-dessous

Président	:	Pr. Olivier ROUX, Professeur	École Centrale de Nantes
Rapporteurs	:	Pr. Alain VIARI, Directeur de recherche INRIA	INRIA Rhône-Alpes
		Dr. Jean-Stéphane VARRÉ, Maître de conférences	Université de Lille
Examinateurs	:	Pr. Olivier ROUX, Professeur	École Centrale de Nantes
		Dr. Laurent NOÉ, Maître de conférences	Université de Lille
Membres invités	:	Pr. Yves VANDENBROUCK, Directeur de recherche CEA	CEA Grenoble

Directeur de thèse : Pr. Irena RUSU
Co-encadrant de thèse : Dr. Damien EVEILLARD
Laboratoire : **L**ABORATOIRE D'**I**NFORMATIQUE DE **N**ANTES **A**TLANTIQUE.
2, rue de la Houssinière, BP 92 208 – 44 322 Nantes, CEDEX 3.

ANALYSE DES SYSTÈMES BACTÉRIENS

UNE APPROCHE *in silico* POUR INTÉGRER LES CONNAISSANCES DU VIVANT

Analysis of bacterial systems

an in silico *approach to integrate life knowledge*

Philippe BORDRON

⋈

favet neptunus eunti

Université de Nantes

Philippe BORDRON

Analyse des systèmes bactériens

une approche* in silico *pour intégrer les connaissances du vivant

x+145 p.

Ce document a été préparé avec LaTeX2e et la classe `these-LINA` version v. 1.30 de l'association de jeunes chercheurs en informatique LOGIN, Université de Nantes. La classe `these-LINA` est disponible à l'adresse :

`http://login.lina.sciences.univ-nantes.fr/`

Impression : these.tex – 8/10/2012 – 11:13

Révision pour la classe : these-LINA.cls,v 1.30 2005/08/16 17:01:41 mancheron Exp

Résumé

L'émergence des expériences dites à haut débit permet l'acquisition rapide de données concernant un système biologique. Les biologistes disposent ainsi, aujourd'hui, d'un nombre important de données de natures hétérogènes qu'ils cherchent à structurer et analyser. Les méthodes dites intégratives proposent de répondre à cette demande, mais la création d'une méthode générale et satisfaisant les requêtes précises des biologistes constitue une tâche ardue.

Ce mémoire s'inscrit dans cette problématique. Nous y abordons diverses méthodes d'intégration des aspects *omiques* (métaboliques, génomiques, transcriptomiques...) d'un système bactérien et nous proposons la nôtre, nommée SIPPER, qui est une méthode générique et flexible. SIPPER permet de retrouver de l'information biologique cohérente entre les différents aspects étudiés grâce à la construction d'un modèle intégratif et l'utilisation d'une distance reposant sur des propriétés ou hypothèses biologiques choisies. Nous avons appliqué SIPPER deux fois sur les données métaboliques et génomiques d'*E. coli*. La première application teste l'hypothèse *les chaînes de réactions successives du réseau métabolique sont catalysées à l'aide d'enzymes produites par des gènes proches sur le génome*, et la seconde teste l'hypothèse *les chaînes de réactions successives sont catalysées par des gènes dont l'expression est similaire*. Nous avons découvert, par ces expériences, des mesures caractérisant certaines entités biologiques comme la *densité génomique* qui permet l'identification d'opérons métaboliques. L'apport de l'intégration de données supplémentaires aux approches n'utilisant traditionnellement qu'un seul type d'information a également été illustré au travers de la génomique comparative. Nous avons ainsi élaboré M&W−IISCS$_{\mathcal{M}}$, une méthode qui calcule des intervalles communs maximaux ayant un fort intérêt *omique*.

Mots-clés : biologie intégrative, informations *omiques*, plus courts chemins, opérons, modules métaboliques, génomique comparative, intervalles de gènes, intervalles communs

Abstract

Nowadays, the emergence of high throughput experiments allows a large number of biological data to be available to biologists, data that they need to structure and analyze. Integrative approaches provide a way to respond to this demand, but the creation of a method that is general and that satisfies the precise requests of biologists is a difficult task.

This is the problem of this thesis. We present various approaches that integrate many *omic* aspects (metabolic, genomic, transcriptomic...) of a bacterial system, and we also propose our own method, called SIPPER, that is both generic and flexible. SIPPER allows us to find consistent biological information between distinct *omic* aspects by constructing an integrated model and using a distance that is based on given biological properties and hypotheses. We apply SIPPER twice on metabolic and genomic data from *E. Coli*. The first application tests the hypothesis that *the chains of successive reactions in a metabolic network are catalyzed by enzymes that are products of neighbours genes in the genome*, and the second application tests the hypothesis that *the chains of successive reactions in a metabolic network are catalyzed by genes that have a similar expression*. These experiments allow us to identify measures that describe some biological entities such as the *genomic density* that allows us to identify metabolic operons. Integrating different kinds of data into traditional approaches that used only one kind of information is also illustrated in comparative genomics. Thus, we have elaborated M&W−IISCS$_{\mathcal{M}}$, a method that computes maximum common intervals that have an important *omic* interest.

Keywords: integrative biology, *omic* information, shortest paths, operons, metabolic modules, comparative genomics, gene intervals, commun intervals

Remerciements

Ces trois années et demie de doctorat furent une formidable expérience durant laquelle j'ai appris beaucoup. La suite de ce manuscrit fait la part belle aux travaux que j'ai réalisés, laissant de côté les aspects humains qui ont rendu possible cet aboutissement. C'est pourquoi, au travers de cette section, je tiens à remercier les nombreuses personnes qui y ont contribué.

Tout d'abord, je remercie vivement ma directrice de thèse, Irena Rusu, et mon co-encadrant, Damien Eveillard de m'avoir proposé ce sujet de travail portant à la fois sur la biologie et l'informatique. Par leur rigueur, leur grande disponibilité, leur patience et leur écoute, ils m'ont permis de m'initier pleinement au travail de chercheur par le biais de discussions, d'encouragements et de conseils.

Ensuite, je remercie chaleureusement Alain Viari et Jean-Stéphane Varré pour avoir accepté d'être rapporteurs de ma thèse, Olivier Roux d'avoir présidé le jury et Laurent Noé d'avoir fait parti des examinateurs. Je remercie aussi Yves Vandenbrouck d'avoir accepté d'être membre invité à ma soutenance de thèse bien qu'il ait eu un empêchement de dernière minute.

Au risque de me répéter, je remercie une nouvelle fois Yves Vandenbrouck et Laurent Noé pour les échanges constructifs réalisés au cours de cette thèse à l'occasion du suivi de ma thèse.

Travailler au LINA sur le campus de l'U.F.R. des Sciences et Techniques de Nantes fut très agréable. Je remercie donc Pierre Cointe, directeur du LINA, Frédéric Benhamou, directeur du département d'informatique, de m'avoir accueilli dans ce laboratoire. Je remercie également tout le personnel administratif pour leur travail qui facilite la vie d'enseignant et de chercheur au quotidien.

Je remercie également les membres présents et passés de l'équipe ComBi que j'ai côtoyé au cours de ma thèse lors de discussions au détour d'un couloir, entre voisin de bureau et au cours de réunions bimensuelles agrémentées de croissants et pains au chocolat. Un grand merci à Sébastien Angibaud, Jérémie Bourdon, Laurent Bulteau, Freddy Cliquet, Damien Eveillard, Guillaume Fertin, Géraldine Jean, Hafedh Mohamed-Babou et Irena Rusu de m'avoir accueilli parmi eux durant ces trois ans et demi de thèse.

Merci aussi à Olivier Roux, Morgan Magnin et Maxime Folschette, membres de l'équipe MeForBio de l'École Centrale de Nantes, de m'avoir accueilli en tant qu'ATER le temps de finir ma thèse et me permettre d'intégrer des notions de dynamique au sein de mon travail.

Je remercie les participants au projet BioTempo, en particulier Anne Siegel, Alejandro Maass, Andres Aravena et Marko Budinich pour la possibilité qu'ils m'ont offerte de travailler sur des données biologiques inédites.

Bien évidemment je ne peux que dire un immense merci à mes compagnons au quotidien durant cette thèse : les doctorants en informatique de l'association Login, et les collègues moniteurs que j'ai pu fréquenter également.

Je remercie grandement, mes parents, ma sœur et mon frère ainsi que le reste de ma famille pour leur soutien et encouragement. Merci également à Alexis, Élodie, Guillaume, Jean et Jean-Marie pour leur amitié.

Enfin et surtout, je remercie profondément Lauriane pour son soutient indéfectible lors de cette dernière année de thèse et avoir transformé ce doctorat en un épanouissement personnel et professionnel.

Sommaire

CHAPITRE 1

Introduction

Afin de comprendre le vivant, l'Homme, au fil de l'histoire, l'a observé de différentes manières. Dans le cadre plus précis de l'étude des organismes bactériens, l'Homme observe, directement ou indirectement, ce qui se passe au sein d'une cellule. De nombreuses observations ont été menées et classées dans divers domaines correspondant chacun à un aspect de la cellule. Par exemple, l'étude de la fonction des protéines constitue la protéomique, l'identification de gènes contenus dans la cellule fait parti de la génétique, tandis que leur l'organisation constitue la génomique, la façon dont les gènes s'expriment est étudiée dans la transcriptomique, l'étude des composés chimiques intervenant dans la cellule désigne quant à elle la métabolomique, tandis que la métabolique étudie comment sont transformés les composés chimiques grâce à des réactions chimiques. Chacun de ces aspects, dit *omique*, décrit ainsi une vue d'un même organisme, le tout constituant alors un système complexe dont certaines portions sont encore inconnues. Dans notre cas, nous nous intéressons aux bactéries, que nous appellons alors systèmes bactériens, et qui sont parmi les plus simples en biologie, mais qui restent malgré tout étonnement complexes. Jusqu'à très récemment, chacun de ces aspects a été traité de façon indépendante des autres, mais la compréhension d'un système vivant dans son ensemble nécessite de les prendre en compte simultanément. Nous assistons donc, depuis quelques années, à l'émergence de méthodes dites *intégratives*, qui utilisent, dans une même analyse, plusieurs aspects *omiques*, afin de comprendre le fonctionnement d'un système vivant.

Quels que soient les aspects considérés, et que ce soit d'un point de vue intégratif ou non, les premières expérimentations sont toujours réalisées sur un faible volume de données. En effet, la vitesse d'obtention des données à propos d'un nouvel aspect est toujours assez lente lors de sa découverte ; mais cela ne dure pas. Un exemple classique est celui du séquençage de génome. Les premiers séquençages demandèrent beaucoup de temps pour aboutir [42, 28, 16]. Cependant, avec l'évolution technique des appareils de mesures, les volumes de données obtenus ont crû de façon importante, et ainsi le nombre de génomes séquencés augmente de façon exponentielle comme l'illustre l'évolution des chiffres disponibles dans [64, 70, 69, 68]. L'automatisation des processus d'analyse de données est l'une des clés de cette amélioration. Cette observation touche directement les méthodes intégratives. Elles sont aujourd'hui confrontées à un problème de passage à l'échelle lié de fait à l'augmentation des volumes de données à disposition, mais aussi au cumul des différents aspects. L'étude manuelle des données est clairement inadaptée pour suivre leur rythme d'obtention, et il devient donc nécessaire de mettre au point des méthodes automatiques et intégratives qui traitent systématiquement ces gros volumes de données.

La problématique de notre sujet se situe là. Nous devons permettre au biologiste de réaliser une étude automatique d'un système bactérien en analysant simultanément plusieurs aspects connus de ce système. Le volume de données utilisées est conséquent ; mais le biologiste, par ses travaux à petite échelle, est capable de décrire le déroulement de l'analyse. Il est aussi capable *a priori* de supposer, en les inter-

polant à partir de ses observations à petite échelle, les résultats attendus à plus grande échelle. Chaque supposition est ainsi la base d'une hypothèse biologique. Il est donc nécessaire de mettre en place un système automatique d'analyse des données à disposition qui indique à quel point une hypothèse formulée est observée sur un volume de données important. L'analyse des résultats obtenus sera supervisée par le biologiste et pourra également être automatisée.

Pour parvenir à cela, il est donc nécessaire, à l'aide des connaissances biologiques disponibles dans la littérature, de créer un modèle biologique structurant cette connaissance. Différentes hypothèses biologiques possibles seront prises en compte dans ce modèle afin de rester le plus générique possible et de s'adapter au maximum de problèmes. À la suite de cette modélisation, une exploration du modèle, à l'aide d'algorithmes de parcours de graphe, sera effectuée en tenant compte de l'hypothèse biologique formulée auparavant afin d'expliquer la connaissance intégrée. Une confrontation des résultats de l'exploration et des observations de la littérature justifiera la validité de l'hypothèse initiale.

Dans ce manuscrit, nous allons tout d'abord étudier plus précisément, dans le chapitre 2, le contexte scientifique dans lequel s'inscrit cette thèse. Nous y présenterons la connaissance biologique considérée, ainsi que les différents aspects d'un système biologique dont nous tiendrons compte. Ensuite nous proposerons dans le chapitre 3 une méthode exploratoire d'analyse biologique nommée `SIPPER`. Elle consiste en la création d'un graphe orienté, jouant le rôle de modèle qui intègre différents aspects biologiques. Dans ce graphe, nous décrirons la recherche automatique de sous-graphes connexes, représentant les parties du système sur lesquelles les aspects biologiques intégrés sont cohérents. Nous présenterons dans le chapitre 4, l'application de `SIPPER` sur les données d'*Escherichia coli*, l'organisme bactérien de référence. Nous y présenterons deux cas de figure. Le premier évalue de quelle façon la réalisation de réactions successives dans un système bactérien est liée à l'expression de gènes voisins dans le génome. Le second évalue à quel point l'enchaînement de réactions successives dans un système bactérien est lié à des gènes s'exprimant de façon similaire. Plusieurs résultats intéressants à propos de l'identification de processus biologiques seront alors observés et discutés. Nous utiliserons par la suite une partie de ces remarques dans le chapitre 5 pour illustrer un apport possible des méthodes intégratives aux domaines plus classiques de la bio-informatique en effectuant de la comparaison de génomes. Nous présenterons en particulier une méthode nommée `M&W-IISCS`$_{\mathcal{M}}$ afin d'apporter des explications à la conservation de groupes de gènes voisins au fil de l'évolution des espèces.

Nous conclurons enfin en rappelant les différents résultats obtenus et en présentant les différentes perspectives de recherche que nous envisageons.

CHAPITRE 2

Contexte scientifique

Pour comprendre le monde, l'Homme se le représente de façon structurée sous la forme de modèles. La compréhension du vivant n'échappe pas à cette règle. Ce chapitre décrit succinctement comment les scientifiques modélisent le vivant et quelles méthodes informatiques ils mettent en place pour analyser ces modèles.

2.1 Description du fonctionnement du vivant

L'une des capacités d'un organisme à se maintenir en vie tient à la manière dont il utilise les ressources à sa disposition via des transformations moléculaires et énergétiques. Ces transformations sont connues sous le nom de *réactions biochimiques.*

Les réactions biochimiques Une réaction biochimique consiste en la transformation de composés chimiques, aussi appelés métabolites, en d'autres composés chimiques. Elle est, par définition, toujours réversible et s'écrit usuellement sous la forme d'une équation (c.f. Figure 2.1(a)). Dans un contexte biologique donné, une réaction est favorisée dans un sens (libération d'énergie sous forme de chaleur, d'électron libre, ...). Pour avoir lieu dans l'autre sens dans le milieu cellulaire elle a alors besoin d'un apport en énergie. Certaines réactions sont difficiles à inverser [29] ; elles sont alors considérées comme irréversibles. Les composés initiaux constituent les *substrats* de la réaction et les composés obtenus sont les *produits* de la réaction. Dans le cas d'une réaction irréversible (c.f. Figure 2.1(b)) les composés qui répondent aux notions de substrats et produits de réaction sont ceux qui sont respectivement les parties gauche et droite de l'équation de la réaction. Dans le cadre d'une réaction réversible, la notion de produit et substrat dépend du sens de la transformation. Lorsque le sens d'une réaction réversible n'est pas précisé, les termes *partie gauche* et *partie droite* de la réaction seront utilisés. Les conditions de réalisation d'une réaction peuvent être assujetties à un processus de catalyse de la réaction. Ce processus est alors réalisé par un type particulier de *protéines* : les *enzymes*.

$$r_{rev} : A + B \rightleftharpoons C + D \qquad r_{irrev} : A + B \longrightarrow C + D$$

(a) Exemple de réaction réversible (b) Exemple de réaction irréversible

Figure 2.1 – Écriture classique des réactions biochimiques qui caractérisent le métabolisme. Les composés A et B sont transformés en composés C et D. Une réaction peut être (a) réversible, avec $\{A, B\}$ la partie gauche de r_{rev} et $\{C, D\}$ la partie droite, ou (b) irréversible, avec $\{A, B\}$ l'ensemble des substrats de r_{irrev} et $\{C, D\}$ l'ensemble des produits.

L'ensemble de ces transformations d'un système biologique constitue son *métabolisme*.

Le métabolisme D'après [1], le métabolisme est l'ensemble des transformations moléculaires et énergétiques (c.-à-d. des réactions biochimiques) qui se déroulent dans une cellule. Ce processus est composé de deux parties distinctes : le *catabolisme* qui représente la dégradation de composés chimiques et qui produit de l'énergie, et *l'anabolisme* qui constitue la synthèse d'éléments organiques grâce à cet apport d'énergie. L'ensemble des réactions intervenant dans le métabolisme peut être synthétisé de façon structurée, sous la forme d'une carte appelée *réseau métabolique*. Cette carte est découpée en fragments appelés *voies métaboliques*. La Figure 2.2, issue de KEGG [56], présente les voies métaboliques du cycle du citrate chez l'ensemble des organismes connus. Les cartes métaboliques constituent un moyen simple et efficace de représenter des processus biologiques. L'étude du métabolisme s'appelle la *métabolique* tandis que l'étude de métabolites qui y participent s'appelle la *métabolomique*.

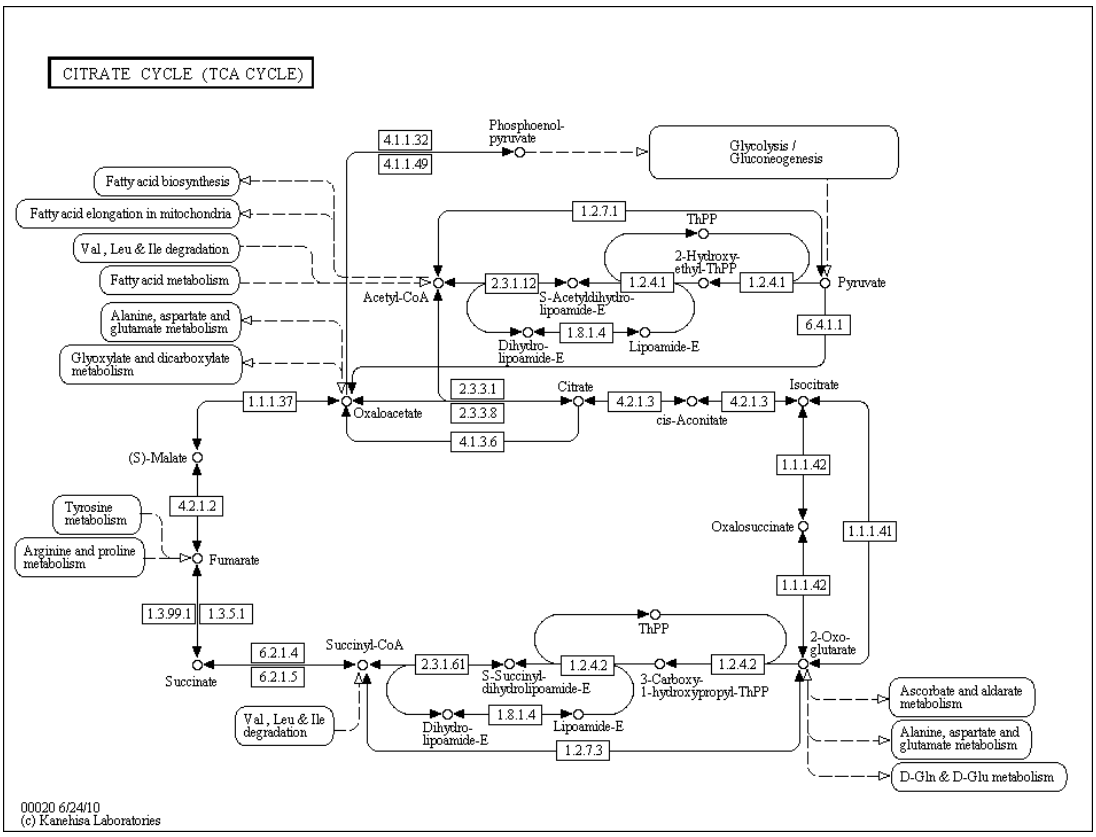


Figure 2.2 – Portion de réseau métabolique issue de KEGG. Cette portion représente les voies métaboliques associées au cycle du citrate pour tous les organismes connus (référence *map00020*). Chaque composé chimique (cercle) est transformé en d'autres par des réactions biochimiques (arcs). Les réactions réversibles possèdent une flèche à chacune de leurs extrémités tandis que les réactions irréversibles en ont seulement à une partie de leurs extrémités. Les étiquettes rectangulaires sur les réactions représentent les enzymes qui catalysent ces réactions. Les rectangles aux coins arrondis représentent d'autres portions du réseau métabolique, et les arcs en traits discontinus entre les composés chimiques et ces rectangles signifient que ces composés chimiques interviennent dans des réactions appartenant à ces portions.

Il existe une multitude de métabolismes à l'échelle des organismes. Comment expliquer une telle diversité ? L'environnement de l'organisme ? Qu'en est-il alors des organismes différents partageant un

même milieu ? Certes, le milieu influe sur le métabolisme de l'organisme, mais la diversité est plutôt à chercher du côté des *gènes* encodés dans les *chromosomes* qui constituent le *génome* de l'organisme [1].

Le génome 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'ADN L'acide désoxyribonucléique ou ADN est la macromolécule constituant le chromosome. Cette molécule est constituée de deux brins complémentaires enroulés en double hélice. Chaque brin est formé d'une séquence d'acides désoxyribonucléiques, molécules appelées aussi 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 (c.f. Figure 2.3). Par abus de langage le terme base désigne souvent le nucléotide lui-même car la base azotée est l'élément qui différencie les nucléotides.

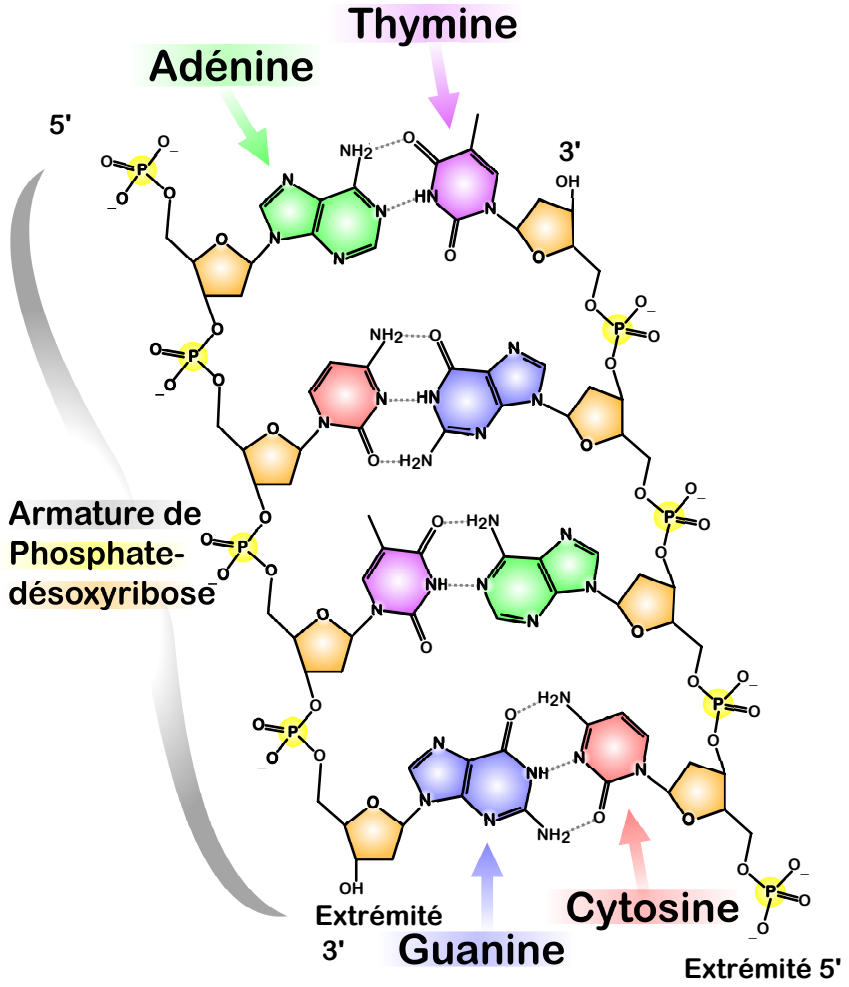


Figure 2.3 – Exemple de la structure de l'ADN. Les deux chaînes de nucléotides sont associées par les liaisons hydrogènes entre leurs bases azotées *(source wikipédia)*.

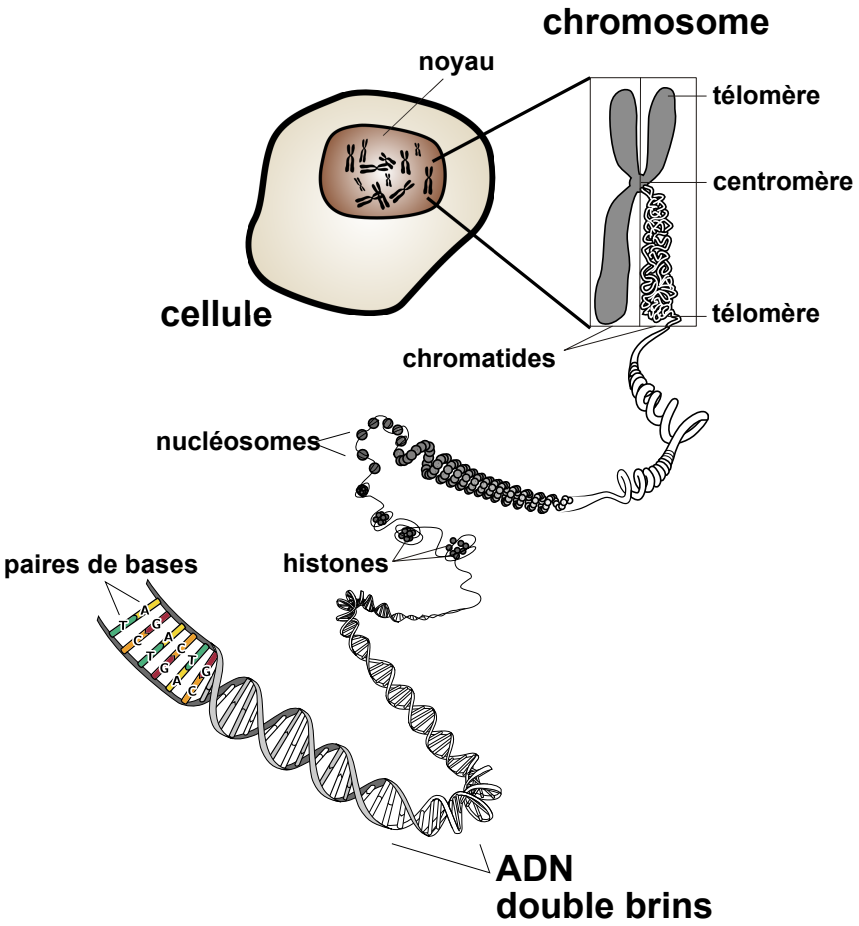


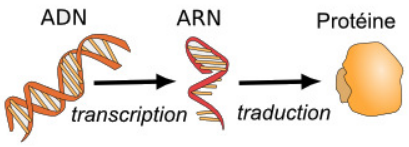
Figure 2.4 – Exemple de la structure d'un chromosome de cellule eucaryote *(source wikipédia)*.

L'ARN L'acide ribonucléique ou ARN est une chaîne simple brin d'acides nucléiques. Chaque acide nucléique est constitué d'un groupement phosphate, un sucre (le ribose) et une base azotée. Sa structure est ainsi très proche de celle de l'ADN, en particulier au niveau des bases azotées qui sont : l'adénine (A), la cytosine (C), la guanine (G) et, à la place de thymine (T) dans l'ADN, l'uracile (U). La complémentarité est conservée : l'adénine s'associe à l'uracile (A=U) et la cytosine s'associe à la guanine (C≡G).

Les 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 constitué de plusieurs millions de ces molécules. Un chromosome est souvent représenté dans la littérature sous sa forme compacte, semblable à un X (c.f. Figure 2.4). 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).

Les gènes Un gène se trouve sur un chromosome. C'est une séquence d'acide désoxyribonucléique (ADN) qui spécifie la synthèse de molécules particulières : les protéines. Cette synthèse se réalise en deux étapes : la transcription puis la traduction.

- La transcription est une étape qui crée une chaîne d'ARN messager (acide ribonucléique) à partir d'une chaîne d'ADN.
- La traduction consiste en la génération d'une *protéine* à partir de l'ARN messager. Cette étape va être effectuée par un ribosome qui va 'lire' le brin d'ARN et reconstituer une chaîne de peptides formant la protéine.



(source wikipédia)

A priori, ces étapes semblent déterministes et produisent une unique protéine chez les procaryotes, mais des régulations *post-transcriptionelles* et *post-traductionnelles* font qu'un gène peut produire plusieurs protéines distinctes. Le(s) produit(s) d'un gène sont donc les différentes protéines que génère le gène suite aux phases de transcription et de traduction. Lorsque ces phases ont lieu, on dit que le gène s'exprime. L'expression d'un gène consiste donc en la production ou la non production de son/ses produit(s). La présence de *facteurs de transcription* en amont d'un gène sur le chromosome permet de moduler l'expression du gène. Par ailleurs, un gène possède un *sens de lecture* qui dépend du brin d'ADN sur lequel il se trouve. Afin de différencier ces sens de lecture, un brin est choisi par convention comme *brin principal* du chromosome et l'autre comme brin complémentaire. Le gène est alors signé afin de préciser le brin sur lequel il se trouve : il porte le signe $+$ s'il est sur le brin principal et le signe $-$ s'il est sur le brin complémentaire. Enfin les différents gènes qui permettent la production d'une même protéine sont dits *homologues*. Dans la cadre de la comparaison d'espèces, deux gènes homologues, chacun appartenant au génome d'une espèce, sont dits *orthologues* s'ils descendent d'un gène unique présent dans le dernier ancêtre commun aux deux espèces. Le science qui étudie les gènes s'appelle la *génétique* et celle qui étudie la transformation des gènes en ARN et protéines s'appelle la *transcriptomique*.

Les 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. Cette fonction est identifiée par l'*E.C. number* dans les bases de données. L'ensemble des protéines générées par un organisme constitue son *protéome*, et la science qui étudie les protéines s'appelle la *protéomique*.

L'indissociabilité du génome et du métabolisme 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 génome et le métabolisme en association constante, l'un et l'autre étant indissociables dans un système vivant.

2.2 L'information métabolique

Le métabolisme caractérise la façon dont les composés chimiques sont utilisés au sein d'un système vivant en fonctionnement. Dans un souhait de synthèse, nous énumèrerons ici les représentations les plus usuelles du métabolisme, ainsi que les analyses informatiques associées.

2.2.1 Description de l'information métabolique à disposition

L'information métabolique consiste en la transformation de composés chimiques, aussi appelés métabolites, en d'autres composé chimiques. L'ensemble de ces transformations est organisé de façon structurée sous la forme d'un réseau métabolique.

Notions de théorie des graphes

La définition d'un réseau métabolique s'appuie essentiellement sur la théorie des graphes. Voici quelques définitions nécessaires à sa compréhension [14, 45].

Définition 1 (Graphe non orienté)**.** *Un graphe non orienté $G = (S, A)$ est défini par un ensemble de sommets S et un ensemble d'arêtes A, chaque arête étant une partie à deux éléments de S. Pour des sommets s_1 et s_2 de S, l'arête allant de s_1 à s_2 est notée s_1s_2 ou s_2s_1, ce qui signifie que $s_1s_2 = s_2s_1$.*

Définition 2 (Graphe orienté)**.** *Un graphe orienté $G = (S, A)$ est défini par un ensemble de sommets S et un ensemble d'arcs A, chaque arc étant un couple ordonné de sommets. Pour des sommets s_1 et s_2 de S, l'arc allant de s_1 à s_2 est noté s_1s_2. Son arc inverse sera noté s_2s_1. Dans le cas orienté $s_1s_2 \neq s_2s_1$.*

Définition 3 (Graphe biparti)**.** *Soit $G = (S, A)$ un graphe non orienté (respectivement orienté [1]) et soient U et V deux sous-ensembles de S. Le graphe G est un graphe biparti si $\{U, V\}$ forme une partition de S telle que chaque arête (respectivement arc) appartenant à A ait une extrémité dans U et l'autre dans V. À des fins pratiques, G sera réécrit de la sorte : $G = (U, V, A)$, avec U et V les parties de S constituant la partition de S et A l'ensemble des arêtes (respectivement arcs) ayant une extrémité dans U et l'autre dans V.*

Définition 4 (Hypergraphe non orienté)**.** *Un hypergraphe non orienté H est un couple $H = (S, A)$ avec S un ensemble fini non vide constituant les sommets de l'hypergraphe et A l'ensemble des hyperarêtes représenté par une famille de parties non vides de S.*

Définition 5 (Hypergraphe orienté)**.** *Un hypergraphe orienté H est un couple $H = (S, A)$, avec S un ensemble fini non vide constituant les sommets de l'hypergraphe et A l'ensemble des hyperarcs : un ensemble de couples ordonnés (X, Y) tels que X et Y soient des parties non vides et disjointes de S. X est alors appelé l'origine de l'hyperarc et Y la destination.*

1. Il n'est pas classique qu'un graphe biparti soit orienté, cependant il est présenté de cette façon dans [39, 107, 82] pour les besoins de la biologie.

Représentation du réseau métabolique

Selon les informations d'intérêt, diverses modélisations du réseau métabolique sont possibles. Il est possible de le représenter sous plusieurs formes : graphe arbritaire, graphe biparti, hypergraphe, et liste de contraintes [65].

Graphe biparti Une des représentations usuelles du réseau métabolique est un graphe biparti (c.f. Figure 2.5(b)). Selon les études, il est orienté [39, 107, 82] ou non [43, 107], voire mixte (composé d'arcs et d'arêtes). Le cas non orienté est utilisé lorsque toutes les réactions sont considérées comme réversibles. Le graphe biparti non orienté $G = (C, R, A)$ est alors défini avec C qui constitue l'ensemble des métabolites qui sont transformés, R l'ensemble des réactions biochimiques, et A l'ensemble des arêtes qui relient les métabolites aux réactions auxquelles ils participent. Ce graphe biparti a une particularité : puisque les réactions transforment des métabolites en d'autres métabolites, le degré d'un sommet appartenant à R est strictement supérieur à 0. L'utilisation de la forme non orientée pose un problème de lecture. Comment déterminer si un métabolite appartient à la partie droite ou la partie gauche de la réaction ? Une solution est alors de considérer la réaction réversible comme deux réactions irréversibles distinctes de sens opposés. Dans ce cas un graphe biparti orienté est plus adapté pour représenter l'orientation des réactions. Les arêtes deviennent des arcs qui caractérisent le rôle de substrat (arc allant d'un métabolite à une réaction) et de produit (arc allant d'une réaction à un métabolite) des métabolites, les réactions réversibles sont codées sous la forme de deux arcs opposés.

Hypergraphe Le réseau métabolique peut être également représenté par un hypergraphe $H = (C, R)$, avec C représentant l'ensemble des métabolites et R l'ensemble des réactions (c.f. Figure 2.5(c)). S'il ne contient que des réactions réversibles, il sera non orienté, et s'il contient des réactions irréversibles, il sera orienté. Dans le cas non orienté, chaque hyperarête $r \in R$ représente une réaction métabolique telle que $r = D \cup G$ avec D l'ensemble des métabolites de la partie droite de la réaction r et G l'ensembles des métabolites de la partie gauche de la réaction r. Dans le cas orienté , chaque hyperarc $r = (D, G) \in R$ représente une réaction irréversible transformant les substrats (ensemble G) en produits (ensemble D).

Graphe de métabolites Le graphe de métabolites est un forme réduite du réseau métabolique. Il se concentre sur les métabolites. C'est un graphe orienté ou non [83], dont les sommets sont les métabolites. Dans le cas non orienté, une arête existe entre deux métabolites s'ils sont respectivement dans la partie droite et dans la partie gauche d'une réaction. Dans le cas orienté, un arc va de chaque substrat à chaque produit pour chacune des réactions (c.f. Figure 2.6(b)).

Graphe de réactions Le graphe de réactions est également une forme réduite du réseau métabolique. L'ensemble des sommets de ce graphe représente l'ensemble des réactions ayant lieu dans le métabolisme. Selon que les réactions sont considérées comme réversibles ou non, le graphe peut-être orienté ou non [71, 31] (voire mixte). Dans le cas non orienté, une arête existe entre deux réactions distinctes si les réactions partagent un métabolite commun. Dans le cas orienté, un arc va d'un sommet r_1 à un sommet r_2 si un produit de la réaction r_1 est un substrat de r_2 (c.f. Figure 2.6(c)).

Graphe d'enzymes Le graphe d'enzymes est une autre forme réduite du réseau métabolique. L'ensemble des sommets du graphe représente l'ensemble des enzymes qui catalysent les réactions du métabolisme. Selon la réversibilité des réactions, le graphe peut-être orienté ou non [105], voir mixte (i.e.

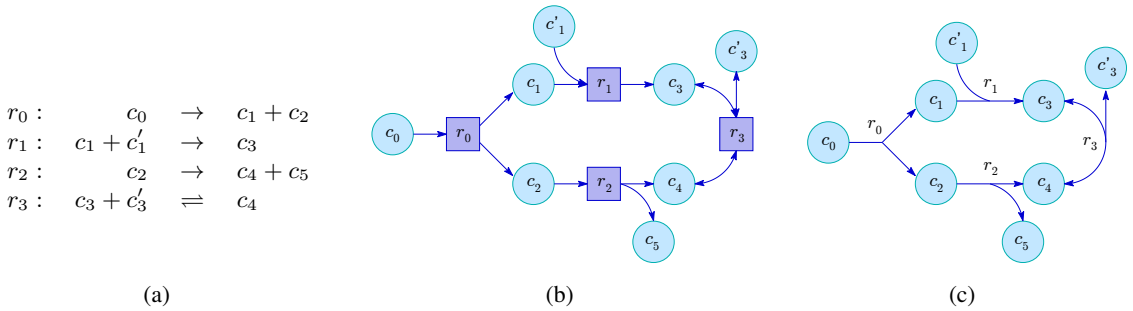


Figure 2.5 – Différentes représentations d'un réseau métabolique. (a) Un ensemble de réactions biochimiques, décrivant le réseau métabolique, peut être modélisé sous la forme de différents types de graphes (ici d'orientation mixte). Le plus souvent le réseau métabolique est représenté sous la forme d'un graphe biparti (b), où les métabolites (cercles) et les réactions biochimiques (carrés) sont les sommets. Lorsqu'un métabolite joue le rôle de substrat d'une réaction, un arc va du métabolite à la réaction, et lorsqu'un métabolite joue le rôle de produit, un arc va de la réaction au métabolite. La représentation sous la forme d'hypergraphe (c) est équivalente à celle sous forme de graphe biparti. Les hyperarcs représentent les réactions (identifiables par leurs labels) qui lient entre eux les métabolites transformés.

composé d'arcs et d'arêtes). Dans le cas non orienté, une arête existe entre deux enzymes distinctes e_1 et e_2 s'il existe une réaction r_1 catalysée par e_1 qui partage au moins un métabolite avec la réaction r_2 qui est catalysée par e_2. Dans le cas orienté, un arc va d'un sommet e_1 à un sommet e_2 s'il existe une réaction r_1 catalysée par e_1 dont un produit est un substrat de r_2 qui est catalysée par e_2 (c.f. Figure 2.6(d)).

Liste de contraintes : Le métabolisme peut aussi être représenté comme une matrice de coefficients stœchiométriques [2] intervenant dans chacune des réactions [37, 94, 97, 80, 66]. Les colonnes de la matrice correspondent aux métabolites intervenant dans le métabolisme, et les lignes aux réactions. Les réactions réversibles ne sont représentables qu'en les dédoublant en deux réactions irréversibles de sens opposé. Lorsqu'un métabolite c joue le rôle de substrat dans une réaction r avec un coefficient stœchiométrique de valeur s, la cellule (r, c) de la matrice prend pour valeur $-s$ (il y a consommation du métabolite). Lorsque le même métabolite c joue le rôle de produit dans la réaction r' avec le coefficient s', la cellule (r', c) de la matrice prend pour valeur s' (il y a production du métabolite). Les métabolites n'intervenant pas dans une réaction r'' prennent pour valeur 0 dans la ligne r''. Cette représentation tient compte de la conservation de la matière dans l'enchaînement des réactions biochimiques (c.f. Figure 2.7).

Problèmes sous-jacents à la modélisation de réseaux métaboliques

La modélisation des réseaux soulève quelques problèmes. En particulier, déterminer si une molécule est importante dans l'enchaînement de réactions [50, 30], est un problème commun à tous les modèles présentés. Par exemple, les molécules d'eau ou d'ATP sont omniprésentes dans le milieu en jouant un rôle auxiliaire au sein d'une majorité de réactions. Ces molécules permettent de modifier des macromolécules et ne constituent pas le métabolite d'intérêt de la réaction. Il n'est donc pas pertinent de les prendre en compte dans toutes les réactions du réseau métabolique, même si, d'un point de vue chimique,

2. Maintenir les proportions stœchiométriques d'une réaction biochimique consiste à obtenir le même nombre d'atomes de même type entre les parties gauche et droite de l'équation de la réaction. Pour cela, chaque composé chimique est pondéré par une valeur entière strictement positive qui est nommée coefficient stœchiométrique.

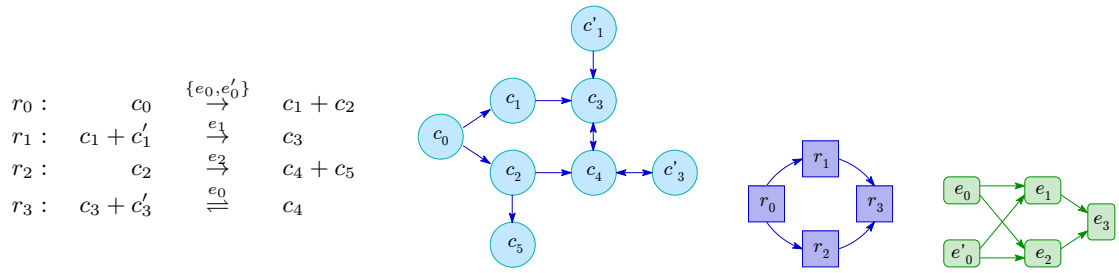


Figure 2.6 – Représentations réduites d'un réseau métabolique. À partir (a) d'un ensemble de réactions, il est possible de centrer la vue du réseau métabolique sur (b) les métabolites par le graphe de métabolites, où chaque métabolite x est relié à un autre y quand x joue le rôle de substrat et y le rôle de produit dans une réaction, ou (c) sur les réactions via le graphe de réactions, qui relie chaque réaction avec une autre quand l'un des produits de la première est le substrat de la seconde, où encore (d) sur les enzymes (graphe d'enzymes) qui se base sur le graphe de réaction en remplaçant chaque réaction par les enzymes qu'ils catalysent.

leur présence est aussi importante que celle de n'importe quelle autre molécule. Il faut donc filtrer de telles molécules en les retirant d'une partie ou de la totalité des réactions. Cela se traduit sur le réseau métabolique soit par le retrait d'arcs, ou de sommets, soit par la mise à zéro de coefficients.

2.2.2 Études des réseaux métaboliques

La décomposition du réseau métabolique en modules

Le découpage, total ou partiel, du réseau métabolique en différents sous-réseaux est important pour en simplifier la compréhension. Chaque sous-réseau constitue alors un processus biologique ayant un rôle fonctionnel. Ce découpage a été en grande partie effectué à la main à partir de la littérature comme le répertorient les bases de connaissances présentées dans [56, 59, 24]. C'est un travail fastidieux qui a entraîné la proposition de méthodes de décomposition et de découpage automatique de celui-ci. Ces décompositions se basent sur la nature topologique du réseau métabolique. Tout d'abord, le réseau métabolique a été perçu comme scale-free [53, 103], ce qui signifie que la distribution du degré des nœuds

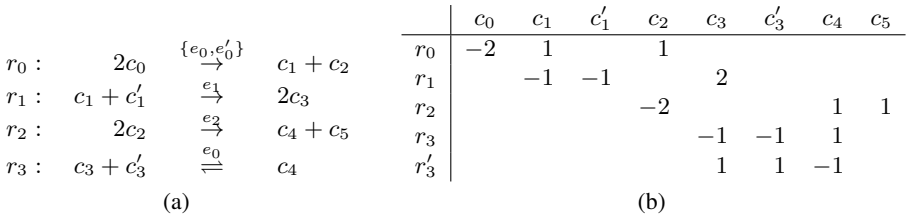


Figure 2.7 – Représentation d'un réseau métabolique sous forme de contraintes. Il est possible de représenter (a) un ensemble de réactions sous la forme (b) d'une matrice. Chaque ligne est une réaction, et chaque colonne un métabolite. Chaque cellule (i, j) représente la quantité de métabolites j consommés (valeur négative) ou produits (valeur positive) dans une réaction i. Les valeurs absentes représentent 0. Les réactions réversibles comme r_3 sont dédoublées en deux réactions irréversibles.

du réseau suit une loi de puissance : en notant par $p(x)$ le nombre de noeuds de degré x, la corrélation entre $p(x)$ et e^{-k} tend vers 1. Cette topologie particulière indiquait que les processus biologiques étaient connectés entre-eux par des points charnières nommés 'hubs'. Cette propriété incitait à regrouper les sommets voisins de même degré ensemble. Cependant, cette propriété était erronée [9] et non compatible avec d'autres propriétés du métabolisme comme sa modularité [96, 103]. Ravasz *et al.* [83] ont proposé une décomposition du réseau métabolique représenté sous la forme d'un graphe de métabolites non orienté, en partant de l'hypothèse que son organisation est hiérarchique. Des sous-réseaux sont cherchés dans le réseau métabolique en utilisant une mesure nommée *coefficient de clustering*, basée sur le degré de connectivité des nœuds voisins, comme critère d'agrégation des nœuds. Les nœuds sont agrégés dans le même cluster quand le coefficient de clustering dépasse un seuil donné, et chaque nœud doit appartenir à un cluster. Les sous-réseaux obtenus sont ensuite contractés en nœuds et de nouveaux clusters sont composés de la même façon. Cette étape est répétée jusqu'à obtenir le cluster contenant tout le réseau métabolique. Cette approche a été généralisée dans [43]. Gagneur *et al.* y proposent un algorithme générique de décomposition hiérarchique sur le réseau métabolique, représenté sous la forme d'un graphe biparti. Leur décomposition permet de regrouper à un même niveau hiérarchique un ensemble de nœuds (nœuds initiaux) et sous-réseaux (nœuds contractés) alors que Ravasz *et al.* associent ensemble soit des nœuds initiaux, soit des nœuds contractés. Cette décomposition, comme celle de Ravasz *et al.* [83], forme une partition du réseau métabolique. Un autre type de clustering a été réalisé dans [71]. Ma *et al.* ont utilisé, dans un graphe orienté de réactions, le plus court chemin entre chaque couple de réactions pour définir une distance entre les réactions métaboliques. Ils ont ensuite utilisé cette distance pour créer la hiérarchie entre réactions. Les clusters sont ensuite calculés en prenant, à partir des racines, les sous arbres de la hiérarchie composés d'au moins 10 nœuds. Une hiérarchie de clusters est obtenue en répétant ainsi cette dernière étape jusqu'à remonter jusqu'à la racine. Il est aussi possible d'obtenir des modules fonctionnels en comparant les réseaux métaboliques de plusieurs espèces. Dans [105], Yamada *et al.* ont mis en place une méthode de clustering basée sur la conservation de la connectivité des enzymes entre espèces sur le graphe des enzymes. De cette façon, les auteurs définissent des modules phylogénétiques du métabolisme. L'idée sous-jacente à cette méthode de décomposition est que si les modules sont conservés, ils le sont pour une raison fonctionnelle.

La recherche de modules topologiques particuliers

Une autre façon d'obtenir de l'information sur les fonctions métaboliques au sein d'un organisme est de rechercher des sous-graphes particuliers, portants le nom de modules, en ce basant sur la structure du réseau, i.e. sa topologie. La recherche de ces sous-graphes permet l'étude d'enchaînements de réactions particuliers. Croes *et al.* [31] ont utilisé, avec $k \in \mathbb{N}$, la recherche des k plus courts chemins entre métabolites dans les 'Small Molecule Metabolism (SMM) network', un réseau métabolique ne s'intéressant qu'à la transformation de petites molécules, pour décrire ces enchaînements. Leur réseau est un graphe biparti orienté et pondéré, où les sommets sont les métabolites et les réactions. Chaque réaction réversible est éclatée en deux réactions irréversibles de sens opposés. Chaque nœud est pondéré par son degré. Ce sont ces poids qui sont pris en compte dans le calcul des k plus courts chemins, avec des contraintes d'incompatibilités entre les sommets qui codent pour une même réaction. Par cette méthode, les auteurs ont calculé les 5 plus courts chemins dans le réseau métabolique entre les extrémités d'une même voie métabolique pour retrouver les voies métaboliques connues, les extrémités jouant alors le rôle d'entrées et sorties des voies métaboliques. Dans la continuité de ces travaux, l'article [39] présente différentes manières de retrouver des voies métaboliques en réalisant de l'extraction de sous-réseaux à partir du réseau métabolique. Pour cela, les auteurs énumèrent plusieurs algorithmes de recherche des k plus courts

chemins et les comparent entre-eux. En analogie avec l'approche décrite dans [105], Yang et Sze [107] ont mis en place des méthodes de recherche de chemins et de sous-graphes les plus similaires (en autorisant des insertions et des suppressions d'éléments biologiques) dans des réseaux biologiques d'espèces différentes, en particulier dans leur réseaux métaboliques. L'algorithme mis en place utilise un chemin ou un sous-graphe en guise de requête pour effectuer cette recherche. Il s'applique sur des graphes orientés ou non, c'est-à-dire qu'il est possible de considérer les réactions biochimiques comme réversibles ou non. Dans l'un des exemples illustrant cet article, les auteurs ont utilisé un réseau métabolique représenté par un graphe orienté biparti (en dupliquant les réactions réversibles en deux réaction irréversibles) pour rechercher des chemins et sous-réseaux conservés entre le réseau métabolique de *E. coli* et celui de *T. thermophilus*.

L'étude des flux sur le réseau métabolique

Un flux dans un réseau métabolique est un ou plusieurs enchaînements de réactions métaboliques permettant la transformation des métabolites en d'autres avec une indication pour chaque métabolite de la quantité de matière relative, qui tient aux proportions stœchiométriques, qui a été générée ou consommée par les réactions.

Dans l'article [37], les auteurs utilisent une méthode nommée FBA (Flux Balance Analysis) afin de tester l'impact de la délétion d'un gène (donc de l'enzyme et réaction associées) sur le fonctionnement de la bactérie *E. coli*. Cette méthode utilise la matrice de coefficients stœchiométriques pour déduire un système d'équations linéaires ayant pour objectif de maximiser la biomasse du système bactérien (c'est-à-dire de maximiser la production d'ATP du système) en fonction des métabolites à disposition dans le milieu et tout en conservant l'équilibre stœchiométrique à l'intérieur du système bactérien.

Dans [94], Schilling *et al.* décrivent un moyen d'analyser les réseaux métaboliques en utilisant le principe de la conservation de la biomasse à l'état d'équilibre du système biologique. Cela correspond aux proportions stœchiométriques qui existent au sein des réactions biochimiques (i.e. la quantité de matière dont sont constitués les substrats d'une réaction est la même que celle dont sont constitués les produits). Il devient alors possible de chercher la façon dont les métabolites initiaux (les entrées du système biologique) sont transformés en métabolites terminaux (les sorties du système biologique). Cela se fait en étudiant la transformation des métabolites dans le réseau métabolique sous la forme de flux équilibré. En utilisant la matrice stœchiométriques $\mathcal{S}$ de taille $n \times m$, avec m le nombre de métabolites et n le nombre de réactions, et un vecteur de flux v de taille n, cela revient à trouve le vecteur v non nul tel que $v.\mathcal{S} = 0$ Dans leur étude, Schilling *et al.* cherchent une décomposition du réseau métabolique en un ensemble élémentaire de flux, appelés modes, qui, par combinaison linéaire de ces modes, reconstitue le réseau métabolique. Ces modes constituent alors des vecteurs générateurs de l'espace vectoriel qu'est le réseau métabolique. Selon les auteurs, les vecteurs générateurs du réseau métabolique forment les bords d'un cône de flux métabolique. Chaque recomposition, même partielle, a pour propriété de maintenir les proportions stœchiométriques à l'état d'équilibre des réactions biologiques qui la composent. L'aspect négatif est que le nombre de modes possibles croît de façon exponentielle en fonction de la taille du réseau métabolique. Pour faire une analyse globale d'un système biologique, il faut donc travailler sur une décomposition du réseau métabolique afin de limiter cette explosion combinatoire [97], mais la plupart des études ne se concentrent que sur une partie de métabolisme [95, 80]

Larhlimi et Bockmayr [66] ont développé une approche similaire à celle des modes élémentaires : la recherche des 'Minimal Metabolic Behaviors' (MMB). À la différence des modes élémentaires, l'espace vectoriel générant le réseau métabolique décrit les faces du cône générateur plutôt que ses bords. Le

cône n'est pas alors nécéssairement pointé, et le nombre de MMB est plus faible que le nombre de modes élémentaires.

L'étude de points d'intérêt du réseau métabolique

Certains travaux sur les réseaux métaboliques se sont concentrés sur les points d'intérêt topologique. Lemke *et al.* [67] expliquent l'impact du retrait d'un métabolite, ou la non-catalyse d'une réaction sur le réseau métabolique, d'un point de vue de la connectivité dans les graphes. Sur des faits similaires, Klampt et Gilles [61] se sont par exemple intéressés aux *ensembles déconnectants minimaux* (*minimal cut sets* en anglais). Un ensemble déconnectant minimal est un ensemble minimal de réactions métaboliques qui, s'il est retiré du réseau métabolique, empêche la production de certains métabolites cibles à partir d'un ensemble de métabolites initiaux. L'étude de ces points d'intérêt révèle la robustesse topologique de certaines voies métaboliques. Pour parvenir à leur fin, les auteurs se sont basés sur la décomposition en modes élémentaires comme support de leurs calculs. Similairement, Rahman et Schomburg [82] définissent d'autres points particuliers du réseau métabolique : les 'load points' [3] et 'choke points' [4]. Un 'load point' est un métabolite incontournable du réseau métabolique. Dans le cadre d'un graphe biparti, c'est un métabolite par lequel passe un grand nombre (par rapport à la moyenne) de k plus courts chemins par rapport au degré de ce métabolite dans le graphe. Un 'choke point' est une réaction (ou enzyme) du réseau métabolique qui consume ou produit un métabolite de façon unique. C'est-à-dire qu'il n'existe pas d'autre réaction (ou enzyme qui catalyse en ensemble de réactions) qui utilise ou génère ce métabolite dans le réseau métabolique. L'importance d'un 'choke point' est évaluée à la fois par rapport au nombre de k plus court chemins passant par ce nœud et par rapport au nombre de 'load points' contenus dans ces chemins. Ces points sont critiques dans le réseau métabolique car les 'load points' caractérisent les métabolites les plus importants topologiquement et les 'choke points' caractérisent les réactions et enzymes qui semblent indispensables topologiquement à certains enchaînements de réactions. L'étude de ces points d'intérêt sert principalement à caractériser la robustesse d'un système vivant face à une carence ou une défaillance de son fonctionnement métabolique [60].

2.3 Informations génomiques

L'étude seule des réseaux métaboliques permet de caractériser le fonctionnement du métabolisme d'un organisme particulier, mais elle n'explique pas pourquoi cet organisme particulier fonctionne de cette façon précise alors que d'autres, dans le même milieu, fonctionnent de façon différente (i.e. ils n'utilisent pas forcément les mêmes composés biochimiques). Certains travaux [37, 99, 48] expliquent *a posteriori* les résultats obtenus à l'aide d'informations issues de la génomique, de la transcriptomique, des interactions de protéines, *etc.* L'idée serait alors de proposer une méthode automatique qui utilise ces informations supplémentaires afin d'apporter des explications (en introduisant une notion de causalité) sur les résultats obtenus. Dans ce manuscrit, les données issues de la génomique et de la transcriptomique seront les sources d'information supplémentaires utilisées afin de fournir des explications à propos du comportement métabolique d'un système bactérien.

3. 'Load point' désigne un 'point de charge' dans le réseau métabolique.
4. 'Choke point' peut être traduit en français par 'goulot d'étranglement'.

2.3.1 Analyse du génome

Le génome est un des éléments qui différencient les systèmes biologiques entre eux. Il semble être la source majeure des différences de comportement observées entre les espèces (particulièrement celles qui évoluent dans le même milieu) et est très utilisé pour les comparer.

Études de l'organisation des génomes

Galperin et Koonin[44] ont étudié l'évolution de la position des gènes homologues sur le génome de différentes espèces au fil de l'évolution. Dans cette étude, les auteurs expliquent qu'il existe des gènes adjacents sur le génome qui codent pour des protéines qui fusionnent (ou fissionnent) ensuite. Ces protéines sont appelées protéines de la pierre de Rossette, en rapport avec le nom de la méthode utilisée pour les identifier. Les auteurs expliquent également que certains gènes adjacents sont responsables de la catalyse de différentes étapes d'une voie métabolique. Ces ensembles de gènes adjacents forment, pour une partie d'entre eux, des opérons [52], c'est-à-dire des groupes de gènes successifs dont l'expression dépend d'éléments appelés *facteurs de transcription*.

En généralisant la notion de voisinage, dans le cas des organismes procaryotes, la proximité des gènes sur le génome semble être très importante. Rocha [87] insiste sur le fait que les gènes proches sur le génome, en particulier les opérons, mais aussi les über-opérons [26] qui sont des groupes d'opérons consécutifs conservés entre organismes, sont responsables d'enchaînements de réactions successifs sur les voies métaboliques. Il précise même qu'une grande majorité d'entre eux sont dans le même ordre que les réactions que leur produits catalysent. Cette observation de Demerec et Hartman [33] a été confirmée expérimentalement par Kovács *et al.* [63].

C'est l'étude de la position relative et de l'ordre des gènes au sein du génome qui sert ainsi de base à la génomique comparative [40]. En effet l'idée sous-jacente à cette approche est que les groupes de gènes conservés ensemble durant le processus d'évolution le sont certainement par nécessité : les gènes conservés partagent souvent un même rôle fonctionnel. Des organismes sont ainsi déclarés plus ou moins semblables en fonction du nombre de similarités et de différences génomiques qui existent entre eux.

Identification des gènes d'intérêt

D'autres études portent non pas sur des ensembles de gènes fonctionnellement liés, mais sur des gènes, qui seuls, ont un rôle important au sein de l'organisme. Il s'agit alors d'évaluer l'impact de la présence d'un gène dans un organisme afin de définir son rôle. Galperin et Koonin [44] ont souligné le fait que certains gènes, à quelques variations près, sont présents dans tous les organismes rencontrés jusqu'alors. Leur présence sur cette multitude d'organismes suggère que certains gènes sont vitaux. Il devient alors intéressant d'identifier ces gènes, dont l'absence rend le système non viable. Ces *gènes essentiels* le sont dans un contexte particulier : celui où la cellule est dans un environnement favorable à sa croissance. Ceux-ci ont été cherchés de façon expérimentale [10]. Dans cet article, Baba *et al.* ont créé des mutants de la bactérie *E. coli* auxquels un gène différent est retiré à chaque fois. Les auteurs ont ensuite constaté la croissance, la non croissance, ou même la disparition de chaque population de mutants de *E. coli* afin de définir si le gène retiré du mutant était essentiel ou non.

L'étude de tels gènes participe à la définition d'un organisme vivant possédant un génome minimaliste, c'est-à-dire composé du plus petit nombre de gènes possibles [46]. Cependant, les gènes essentiels seuls ne suffissent pas pour définir un tel organisme [38, 46].

2.3.2 Expression du génome

Si certains gènes, comme les gènes essentiels, semblent plus importants que d'autres, la position et l'ordre des gènes sur le génome ont un rôle certain dans la réalisation de fonctions biologiques. Toutefois, l'étude de l'expression des gènes montre qu'il existe des liens entre gènes qui ne dépendent pas de ces propriétés. En effet, si des gènes s'expriment de façon similaire dans un contexte particulier, par rapport au fil du temps ou entre espèces ou mutants, c'est que ces gènes sont probablement liés par un mécanisme. Les gènes sont alors *coexprimés*. Michalak [74] présente différentes justifications à la coexpression des gènes au sein d'un génome : soit les gènes sont exprimés de façon similaire dans des espèces comparées (i.e. raison évolutionnaire), soit l'expression de plusieurs gènes est régulée par un facteur de transcription commun (i.e. raison régulatoire), soit les gènes sont proches sur le génome (i.e. raison de colocalisation), soit les produits de gènes participent aux mêmes voies métaboliques, ou soit leurs protéines interagissent ensemble...

La recherche de la coexpression de gènes se réalise en plusieurs étapes décrites précisément dans les sous-sections suivantes qui s'articulent de la façon suivante. Elles consistent tout d'abord en un ensemble de mesures de l'activité des gènes obtenues grâce aux puces à ADN. Une similarité d'expression des gènes deux à deux est ensuite établie, et il est enfin possible de définir des groupes de gènes dont les niveaux d'expression sont fortement corrélés.

Les puces à ADN

L'expression d'un gène se mesure à partir de la quantité de produits du gène (ARN ou protéines) présent dans l'échantillon prélevé à un instant t donné. En faisant des relevés à des moments réguliers, il est possible de suivre la dynamique d'expression du gène en fonction du temps. Il est également possible d'établir la dynamique d'expression du gène en fonction de la variation de paramètres autres que le temps, comme la variation du pH ou de la concentration d'un produit particulier dans le milieu cellulaire... Les informations d'expression des gènes sont obtenues par des puces à ADN (voir Figure 2.8).

La similarité d'expression

À partir des valeurs d'expression de chaque gène, il est possible de définir la similarité d'expression de deux gènes. Stuart *et al.* [100] utilisent comme mesure de similarité s le coefficient de corrélation de Pearson entre les niveaux d'expression de chaque couple de gènes (g_1, g_2) :

$$s(g_1, g_2) = |cor(g_1, g_2)| \text{ avec } cor(g_1, g_2) \text{ la corrélation d'expression de } g_1 \text{ et } g_2 \text{ et } 0 \leq s \leq 1$$

Cette mesure est modulable par des fonctions mathématiques afin d'amplifier les fortes valeurs de corrélation d'expression et de réduire les faibles valeurs [109]. Il est aussi possible d'utiliser les valeurs d'intensité moyennes pour définir la similarité. Grâce à l'utilisation d'un de ces scores de similarité, une matrice de similarité d'expression entre les gènes est générée. C'est à partir de celle-ci que les groupes de gènes coexprimés (i.e. ceux qui sont suffisamment similaires au niveau de leur expression) vont être recherchés.

En appliquant la fonction de similarité d'expression s à chaque couple de gènes, il est possible de définir une matrice de similarité d'expressions de gènes S, où chaque cellule $S[i, j]$ correspond au score de similarité d'un couple de gènes (g_i, g_j). Notons que S est une matrice triangulaire (voir Figure 2.9(a)). À partir de la matrice de similarité S, il est alors possible de définir, à l'aide d'une valeur seuil donnée, une matrice d'adjacence de gènes A. Cette matrice décrit alors quels sont les gènes coexprimés ou non (voir

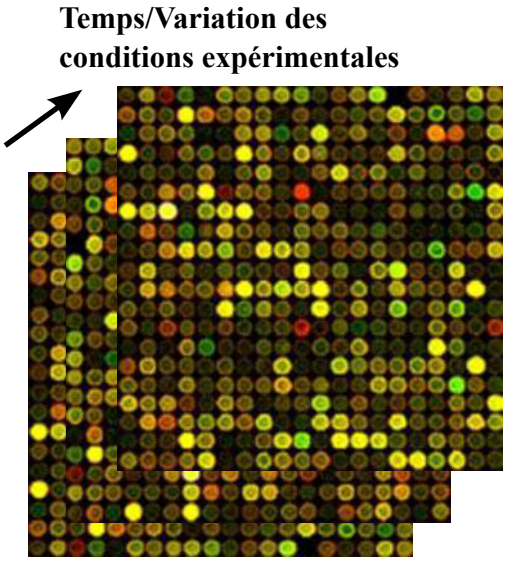


Figure 2.8 – Exemple de puces à ADN. Pour chaque puce l'expression d'un ensemble de gènes est mesurée grâce à l'hybridation de son ARN avec un brin d'ADN complémentaire qui a des propriétés fluorochromes. Plusieurs relevés sont effectués en fonction des informations qui varient (conditions expérimentales, temps, ...) et permettent ainsi d'obtenir des profils d'expression de gènes.

Figure 2.9(b)). Une autre façon de représenter la matrice d'adjacence est de dessiner un graphe de coexpression. Les sommets représentant alors les gènes et les arcs encodent l'existence d'une coexpression entre les gènes à leurs extrémités (voir Figure 2.9(c)).

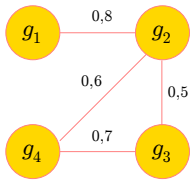
Groupes de gènes d'expression corrélée

Pour obtenir des groupes de gènes coexprimés, il est possible de comparer plusieurs espèces ensemble. En utilisant les données obtenues de 3182 puces à ADN portant sur le génome de l'homme, de la mouche, du vers et de la levure, Stuart *et al.* [100] ont étudié quels pouvaient être les modules métaboliques conservés afin d'identifier l'interaction des gènes conservés d'un point de vue évolutionnaire. Pour cela, les auteurs ont calculé tout d'abord les gènes orthologues entre les génomes avec BLASTp [89], un algorithme de comparaison de séquences de protéines, pour obtenir ce qu'ils appellent des metagènes (un gène partagé entre plusieurs organismes est un métagène). Chaque gène d'un génome est associé au plus à un métagène. Ensuite, les auteurs recherchent les groupes de métagènes dont l'expression est corrélée sur plusieurs expériences (i.e. les conditions de stress des expériences diffèrent) parmi plusieurs organismes. Un classement de ces corrélations est effectué et la *p-value*[5] d'observer par chance un tel classement entre les différents organismes est calculée afin de déterminer si la corrélation observée pour chaque groupe n'est pas le fruit du hasard. Si un groupe de métagènes est assez significatif (par rapport à sa *p-value*), alors le groupe forme un ensemble de métagènes coexprimés qui codent pour une fonction conservée entre les espèces. De façon plus similaire à la décomposition de réseaux métaboliques, Zhang et Sze [109] proposent une décomposition hiérarchique des réseaux de coexpression basée sur la

5. La *p-value* est une mesure de qualité d'un résultat par rapport à l'aléatoire. C'est-à-dire que si une valeur observée sur un résultat est plus petite que la *p-value* calculée, alors le résultat n'est pas dû au hasard. Une définition exacte est disponible dans la section 4.4 de cette thèse.

$$S = \begin{array}{c} \\ g_1 \\ g_2 \\ g_3 \\ g_4 \end{array} \begin{array}{c} \begin{array}{cccc} g_1 & g_2 & g_3 & g_4 \end{array} \\ \begin{pmatrix} - & 0,8 & 0,2 & 0,3 \\ & - & 0,5 & 0,6 \\ & & - & 0,7 \\ & & & - \end{pmatrix} \end{array} \qquad A = \begin{array}{c} \\ g_1 \\ g_2 \\ g_3 \\ g_4 \end{array} \begin{array}{c} \begin{array}{cccc} g_1 & g_2 & g_3 & g_4 \end{array} \\ \begin{pmatrix} - & 1 & 0 & 0 \\ & - & 1 & 1 \\ & & - & 1 \\ & & & - \end{pmatrix} \end{array}$$

(a) Matrice de similarité (b) Matrice d'adjacence (seuil$\geq 0,5$)



(c) Graphe de coexpression

Figure 2.9 – De la corrélation d'expression de gènes au graphe de coexpression. À partir (a) d'une matrice de similarité d'expression S, (b) une matrice d'adjacence des gènes A est calculée : pour chaque valeur supérieure ou égale à un seuil de $0,5$, les gènes sont considérés comme adjacents (valeur 1) ou non (valeur 0). La matrice d'adjacence peut alors être représentée sous la forme (c) d'un graphe non orienté, dont les sommets sont les gènes et dont les arcs représentent la relation d'adjacence de A (la valeur de similarité entre gènes a été ajoutée aux arcs à titre indicatif).

méthode de Ravasz *et al.* [83]. Les auteurs génèrent d'abord une matrice d'adjacence de gènes à partir de la matrice de similarité d'expression de gènes : si la valeur d'une cellule de la matrice de similarité est supérieure à une valeur seuil, alors les gènes sont adjacents (i.e. coexprimés). La matrice d'adjacence encode alors pour un graphe de coexpression (c.f. Figure 2.9). Il sera ensuite décomposé avec la méthode mise en place par Ravasz *et al.* (voir Section 2.2.2), mais avec une prise en compte du poids des arcs, qui correspond à la similarité d'expression des gènes deux à deux. Les auteurs ont illustré leur méthode dans le cadre d'une étude sur le cancer du cerveau et du cycle cellulaire de la levure. Cette méthode fait d'ailleurs apparaître qu'il existe une forte corrélation entre la connectivité des gènes dans le réseau de coexpression, et leur essentialité [23]. La limite étant toujours de déterminer le seuil de corrélation à partir duquel il faut considérer les gènes comme coexprimés ou non. Ruan *et al.* [88] ont proposé une méthode nommée `QCut` pour s'affranchir de ce problème de seuil. `QCut` est une méthode qui génère de façon automatique des partitions du graphe de coexpression. Elle repose sur une fonction de modularité afin de déterminer si chaque partition est due au hasard ou non. La partition de meilleur score est sélectionnée comme décomposition du graphe de coexpression. Ainsi, il n'y a pas besoin de définir de seuil particulier, ni un nombre de modules, mais il faut générer plusieurs candidats *a priori* intéressants. Dans cet article, les auteurs utilisent un algorithme standard de découpage spectral appliqué récursivement sur chaque partie des partitions obtenues jusqu'à ne plus pouvoir améliorer le score de la fonction de modularité.

Le problème majeur de l'étude des graphes de coexpression est de fixer une limite à partir de laquelle des gènes sont coexprimés ou non. En effet, en fonction de l'expérience, ce seuil varie. Une méthode admise est de chercher si la valeur de coexpression obtenue est due au hasard ou non. L'association des données de coexpression avec d'autres informations *omiques* (génomiques, métaboliques, ...) comme celles proposées par Michalak dans [74] pourrait aider à s'affranchir de cette limite.

2.4 Intégration des données *omiques*

Afin de comprendre le système vivant dans sa totalité, il est nécessaire de prendre en compte chacune des informations le concernant, c'est-à-dire d'utiliser les données *omiques* ensemble. Certaines des différentes approches qui suivent sont dédiées à une question biologique précise, tandis que d'autres sont plus générales et intègrent des informations de différentes natures à des fins exploratoires.

2.4.1 Méthodes dédiées utilisant l'intégration de données *omiques*

Rison *et al.* [85] ont été parmi les premiers à travailler selon cette démarche. En effet, ils ont cherché à connaître quelles relations existent entre différentes caractéristiques biologiques : le nombre de réactions catalysées entre deux réactions ou enzymes dans le réseau métabolique, l'homologie des enzymes et la position de leur(s) gène(s) sur le génome. Pour cela, les auteurs ont évalué différentes propriétés biologiques les unes par rapport aux autres en étudiant leurs corrélations et leurs distributions. Les auteurs se sont concentrés plus particulièrement sur les notions de chaîne de réactions, de taille d'intervalle de gènes, mais aussi de pourcentage de gènes homologues. Il en résulte que (a) les enzymes qui participent à des mêmes voies métaboliques semblent être encodées par des gènes proches sur le génome, en particulier les opérons, (b) que les enzymes qui sont proches sur le réseau métabolique tendent à être plus homologues que les enzymes éloignées, (c) que les gènes proches sur le génome tendent plus à être homologues que les gènes distants. Les auteurs estiment cependant que les points (b) et (c) sont mitigés car il n'y a pas assez de cas de figures pour en être certain. Ces travaux ont été repris et retravaillés dans [98] en étudiant plus particulièrement, à l'aide d'un algorithme de calcul du plus court chemin métabolique, les notions de proximité génomique et enchaînement de réactions métaboliques, mais aussi sur le classement fonctionnel des enzymes. Il ressort de cette étude qu'il existe une forte corrélation entre les gènes proches sur le génomes et les réactions successives sur le réseau métabolique, mais aucune corrélation significative entre la fonction des enzymes et l'enchaînement de réactions. Hattori *et al.* [48] ont également fait une approche similaire afin d'évaluer le lien entre la proximité des gènes sur le génome et la proximité des réactions sur le réseau métabolique.

En utilisant le même constat que les études précédentes, Zheng *et al.* [110] ont déterminé des clusters de gènes proches à la fois sur le réseau métabolique et sur le génome afin de prédire des opérons. Pour cela les auteurs ont utilisé, d'une part, le réseau métabolique comme un graphe biparti non orienté où les sommets sont des métabolites ou des enzymes, et d'autre part, le génome comme un graphe non orienté dont les sommets sont les gènes et dont les arcs représentent l'adjacence de deux gènes. Un cluster de gènes est obtenu en calculant tout d'abord, à partir d'une enzyme particulière, nommée la racine, tous les sommets accessibles dans un périmètre de δ_m nœuds, soit un arbre de profondeur δ_m. Ensuite les sommets obtenus sont projetés sur le génome grâce à la relation 'une enzyme est un produit de gène' (via leur EC number). Les gènes trop distants ($> \delta_g$) des autres gènes centrés sur la racine sont retirés du cluster de gènes.

Dans un autre registre, Joyce *et al.* [54] ont testé l'essentialité des gènes d'*E coli* en simulant l'impact de leur retrait du génome aussi bien sur le réseau métabolique, via les enzymes qu'ils catalysent, que sur la transcription des autres gènes, en invalidant leur action sur les facteurs de transcription. Pour parvenir à leurs fins, les auteurs ont créé des génomes et réseaux métaboliques mutants. À chaque fois ils ont supprimé un gène du génome original, et ont propagé cette suppression au réseau métabolique en retirant les réactions catalysées uniquement par les produits de ce gène. Une simulation de la croissance des mutants a été effectuée par une FBA. Les métabolites d'entrée et de sortie du réseau métabolique correspondent à ceux utilisés dans les médiums de [10].

De façon similaire, afin de comprendre l'adaptation de *E. coli* à des perturbations environnementales, Ishii *et al.* [51] ont utilisé un mélange de méthodes expérimentales et de simulations informatiques. Les auteurs ont ainsi mesuré la réponse globale de *E. coli* à différents niveaux (expression des protéines et des gènes) et ont comparé celle-ci à une analyse quantitative des voies métaboliques spécifiques. Grâce aux informations connues à propos des gènes essentiels, répertoriées dans la base KEIO [10], et associées à une analyse des flux métaboliques sur le réseau métabolique de *E. coli*, les auteurs ont confirmé que les modèles informatiques existants pouvaient simuler précisément les perturbations introduites sur ces dits

modèles. Cependant, beaucoup d'informations sont nécessaires pour parvenir à une telle simulation sur un organisme donné.

Notebaart *et al.* [77] ont étudié quelle information issue du réseau métabolique (proximité des réactions ou flux) explique le mieux la dynamique d'expression des gènes (i.e. la corégulation). Il résulte de cette étude que les analyses de flux tendent à mieux justifier pourquoi les gènes sont coexprimés, en particulier car les gènes couplés par les analyses flux (i.e. FBA) montrent des profils d'expression similaires, mais partagent aussi des régulateurs communs et résident fréquemment dans le même opéron.

2.4.2 Méthodes générales d'intégration

Des méthodes générales de traitement des information *omiques* ont déjà été mise en place. Par exemple, afin de déterminer des clusters d'enzymes fonctionnellement liées (FRECs), Ogata *et al.* [78] ont choisi de comparer des données biologiques de natures différentes. Les auteurs ont utilisé le réseau métabolique et le génome. Le réseau métabolique est représenté sous la forme d'un graphe d'enzymes non orienté tandis que le génome comme graphe non orienté également dont les sommets sont les gènes et les arêtes lient deux gènes adjacents dans le génome. Dans le but d'obtenir des modules de natures différentes, les deux graphes ont été comparés. Une correspondance à été établie entre les sommets des deux graphes par les E.C. numbers associés à chaque gène et enzyme. Un sommet associé à plusieurs autres dans le graphe opposé est dupliqué. Au départ du découpage, tous les sommets sont dans des modules différents. Ensuite deux modules c_1 et c_2 sont fusionnés si, dans chacun des graphes, il existe un chemin entre les sommets de c_1 et ceux de c_2 dont le poids est inférieur à un seuil donné. Les modules sont ainsi fusionnés tant qu'il est possible de les fusionner.

Nakaya *et al.* [76] ont généralisé les travaux de Ogata *et al.* à plus de deux graphes en élaborant une méthode de recherche de clusters de gènes corrélés au niveau du génome, des voies métaboliques, de la similarité de coexpression. Il proposent aussi de réutiliser cette méthode afin de comparer des génomes par rapport à leurs graphes de coexpression de gènes.

Le défaut des travaux de Ogata *et al.* ou Nakaya *et al.* est qu'ils ne s'intéressent qu'à un type information particulier (les enzymes ou les gènes) alors qu'ils ont intégré plusieurs aspects d'un même système bactérien. À l'opposé de cette démarche, Boyer *et al.* [20] ont considéré que, plutôt que d'en extraire un type particulier, toute l'information biologique qui a été obtenue en résultat sous sa forme intégrée est intéressante. Ils ont ainsi déterminé des modules de gènes, enzymes et réactions biochimiques associés dans un possible même processus biologique en élaborant une méthode basée sur l'analyse d'un multigraphe. Le multigraphe étudié consiste alors en la superposition de différents graphes biologiques. Chaque sommet du multigraphe est alors un n-uplet de gènes, enzymes et réactions ; chaque élément du n-uplet étant un sommet dans un des graphes superposés. La relation qui les associe est : 'les réactions sont catalysées par des enzymes qui sont des produits de gènes'. Chaque arc encode alors la relation entre deux sommets : gènes voisins, gènes coexprimés, réactions successives, interaction d'enzymes. Par la suite, les auteurs cherchent des composantes connexes communes dans ce multigraphe. Chaque composante connexe commune représente un sous-graphe connexe dans chacun des graphes qui composent le multigraphe étudié. Cette approche permet d'obtenir des modules à la fois métaboliques, génomiques, enzymatiques...

2.5 Le travail de cette thèse

La méthode développée dans cette thèse, nommée `SIPPER`, cherche à concilier les deux familles de méthodes évoquées ci-dessus, c'est-à-dire à développer une méthode générale qui recherche des enchaî-

nements de réactions sans *a priori* et qui s'adapte bien à des cas spécialisés. Le chapitre 3 présentera donc SIPPER d'un point de vue général et le chapitre 4 en présentera des applications dédiées sur *Escherichia coli*. Certains résultats observés lors de cette application seront ensuite réutilisés dans le chapitre 5 pour mettre en place une méthode de comparaison de génomes, nommée M&W-IISCS$_{\mathcal{M}}$, intégrant de l'information *omique*.

CHAPITRE 3

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

Ce chapitre présente de façon détaillée les travaux publiés dans [18]. Dans ce chapitre, nous allons décrire notre méthode SIPPER. Nous allons introduire la notion de modèle intégré $\mathcal{G}_{int}$, un graphe cumulant des informations biologiques de différentes natures, dans lequel nous allons chercher des sous-graphes particuliers, nommés k-SIPs. À chaque étape, nous présenterons la motivation biologique associée et les difficultés sous-jacentes liées aux problèmes théoriques entrant en jeu. Nous présenterons aussi comment utiliser les particularités de $\mathcal{G}_{int}$ pour contourner une partie d'entre eux. Nous comparerons enfin SIPPER à d'autres méthodes existantes présentées dans le chapitre précédent.

3.1 Introduction

Le chapitre précédent fait ressortir le point clé suivant : l'organisation, et par conséquent l'analyse de l'information biologique, est basée essentiellement sur la notion d'adjacence [44] et surtout sa généralisation, le concept de connectivité [20]. Lorsque l'information biologique est organisée sous la forme d'un réseau, comme le décrivent certains exemples énoncés dans le chapitre 2 [85, 20, 100], la connectivité exploite les adjacences successives. Ces dernières sont faciles à manipuler dans les réseaux, ce qui permet de traiter l'information potentiellement manquante (arc ou arête), tant qu'il existe une manière alternative de relier des sommets ensemble. L'intégration de différents types d'information doit ainsi être basée sur la formalisation de ces concepts d'adjacence et de connectivité, comme suggéré dans [20]. La théorie des graphes offre à la fois le degré d'abstraction et certains outils nécessaires à une telle approche. Cependant, toutes les notions de voisinage issues de la théorie des graphes ne sont pas biologiquement significatives [77]. Il est alors important de définir une méthode appropriée pour traiter la notion de proximité au sein de données biologiques hétérogènes. Les études présentées dans la Section 2.4 montrent l'importance de l'analyse simultanée des différents aspects d'un système biologique afin d'en comprendre le comportement. L'exemple le plus remarquable est celui mis en avant dans [44], qui analyse différents points de vue de la proximité des gènes comme la colinéarité des gènes, les profils de co-expression, le domaine de fusion des protéines (protéines de la pierre de Rosette) ou les profils phylogénétiques.

Devant l'extrême abondance et la diversité des cas présentés dans du chapitre 2, nous proposons ici une méthode, nommée SIPPER [18], dont le but est de fournir un cadre formel à une grande partie de ces approches, tant qu'un réseau métabolique entre en jeu. Ceci est réalisé par :

1. l'intégration, dans les systèmes bactériens, de l'information de voisinage/non-voisinage à propos des protéines et des gènes (provenant des analyses de séquence/structure) dans le réseau métabolique via le paradigme *un gène produit des enzymes*,
2. l'extension, à une plus grande échelle, de l'analyse locale basée sur le voisinage en utilisant l'analyse de la connectivité au sein d'un réseau,
3. la possibilité de tenir compte des informations de proximité entre gènes indépendamment des données par une distance librement choisie.

Notre méthode SIPPER nous permet alors d'étudier simultanément, de façon flexible et automatique (a) la proximité relative de composants métaboliques dans le réseau métabolique (b) en accord avec une notion appropriée de distance portant sur un ou plusieurs aspects *omiques* du système bactérien. Par la notion de distance, nous entendons la définition suivante :

Définition 6 (distance)**.** *La distance d sur un ensemble E est une fonction de $E \times E$ dans $\mathbb{R}^+$. Elle respecte les conditions suivantes :*
symétrie : $\forall x, y \in E, d(x, y) = d(y, x)$
séparation : $\forall x, y \in d(x, y) = 0 \Leftrightarrow x = y$

Cette définition de distance diffère de celle usuellement rencontrée dans la littérature, qui contient la condition supplémentaire d'inégalité triangulaire. Lorsqu'une distance d vérifie cette condition, nous l'appellerons métrique.

Définition 7 (métrique)**.** *Le distance d sur un ensemble E est une métrique si elle vérifie la condition suivante :*
inégalité triangulaire : $\forall x, y, z \in E, d(x, z) \leq d(x, y) + d(y, z)$

Des exemples possibles de distances entre gènes sont : la distance intergénique [22, 27], la distance de colocalisation [13], la mesure de similarité d'expression (voir Section 2.3.2, [100]). Des exemples possibles de distances entre enzymes sont : le plus court chemin entre les enzymes dans le PPI ou n'importe quelle distance fonctionnelle entre les enzymes [98]. Ces listes ne sont pas exhaustives, puisque n'importe quelle distance alternative ou combinaison de distances est également utilisable dans notre méthode.

La fonctionnement de SIPPER consiste en trois étapes : (A) construction du modèle intégratif $\mathcal{G}_{int}$ avec une hypothèse biologique sous-jacente (pondération des arcs), (B) recherche des k-SIPs, k chemins minimisant une distance biologique dans $\mathcal{G}_{int}$, (C) sélection des chemins d'intérêt parmi ces k-SIPs.

3.2 Construction d'un modèle intégrant métabolisme et génome

Dans le chapitre précédent, nous avons vu que le réseau métabolique pouvait se représenter sous la forme de graphes. Bien que cette représentation provoque la perte de l'information stœchiométrique des réactions, elle permet néanmoins des analyses donnant des résultats biologiquement cohérents. Ces représentations sont cependant limitées aux informations métaboliques. L'intégration d'information génomique ou enzymatique au réseau métabolique nécessite alors la définition d'une nouvelle représentation. La représentation que nous proposons sera appelée *modèle intégré*, et le graphe pondéré résultant sera noté $\mathcal{G}_{int}$.

3.2.1 Informations nécessaires à l'élaboration du modèle intégré

Nous allons nous intéresser aux réactions dont l'enchaînement est guidé par la proximité des gènes qui les catalysent via leurs enzymes. La conception du modèle intégré se base donc sur la relation qui existe entre les gènes, les enzymes et les réactions (Section 2.1) : un gène produit des protéines qui peuvent jouer le rôle d'enzymes en catalysant des réactions biochimiques. Afin de parvenir à notre modèle biologique, nous considérons alors le réseau métabolique $\mathcal{M}$ comme un ensemble de réactions. Chaque réaction transforme des substrats en produits avec la particularité que les métabolites présents dans un trop grand nombre de réactions, comme l'eau ou l'ATP, sont ignorés dans les réactions [50, 30]. Une réaction de $\mathcal{M}$ peut-être réversible ou non. Nous considérons également le génome $\mathcal{G}$ d'un organisme unichromosomal comme une séquence circulaire ou linéaire de gènes (voir Figure 3.1(a)). Chaque gène de $\mathcal{G}$ possède donc deux voisins (sauf aux extrémités, dans le cas linéaire). Les produits d'un gène g donné de $\mathcal{G}$, des protéines, peuvent jouer le rôle d'enzymes dans $\mathcal{M}$. La fonction *enzymes*(g) associe à chaque gène g l'ensemble des enzymes qui sont produites par g et qui catalysent des réactions de $\mathcal{M}$. Comme les enzymes obtenues catalysent des réactions, il est aussi intéressant de connaître quelles sont les réactions qui sont catalysées par une enzyme donnée. La fonction *reactions*(c) va dans ce sens. Elle associe à chaque enzyme (ou gène respectivement) c l'ensemble des réactions de $\mathcal{M}$ catalysées par c (respectivement par les enzymes de *enzymes*(c)). Comme nous travaillons soit sur les gènes, soit sur les enzymes, nous fixons alors l'ensemble des *unités catalytiques* $\mathcal{C}$ comme étant soit l'ensemble des enzymes qui catalysent les réactions de $\mathcal{M}$, soit l'ensemble des gènes qui produisent des enzymes qui catalysent les réactions de $\mathcal{M}$. Ce choix permet de distinguer deux versions de SIPPER : la version "gène" et la version "enzyme".

Dans la réalité du vivant, la transcription d'un gène en protéine n'est pas aussi immédiate. En effet, un gène est transcrit en un polypeptide qui est composé d'acides aminés, puis ce polypeptide peut prendre la forme de différentes protéines (et donc enzymes) lorsqu'il s'associe à d'autres polypeptides. Ils forment ainsi ensemble des complexes polypeptidiques qui sont aussi appelés protéines. Ainsi une chaîne polypeptidique peut potentiellement catalyser plusieurs réactions. Dans notre cas, une protéine ne sera que la chaîne polypeptidique.

3.2.2 Le modèle intégré : $\mathcal{G}_{int}$

Étant donnés un ensemble d'unités catalytiques $\mathcal{C}$ et une distance d entre les unités catalytiques de $\mathcal{C}$, le modèle intégré $\mathcal{G}_{int} = (V, E)$ est un graphe orienté défini comme suit.

Sommets de $\mathcal{G}_{int}$ L'ensemble des sommets V de $\mathcal{G}_{int}$ est formé de toutes les paires (unité catalytique c, réaction r) telles que r appartient à *reactions*(c). De façon formelle nous avons :

$$V = \{(c, r) \,|\, c \in \mathcal{C}, r \in \mathcal{M}, r \in \textit{reactions}(c)\}$$

Arcs de $\mathcal{G}_{int}$ Un arc de E va d'un sommet (c_x, r_x) vers un sommet (c_y, r_y) quand l'un des produits de r_x est un substrat de r_y, avec $r_x \neq r_y$. De façon formelle nous avons :

$$E = \{(x, y) \,|\, x = (c_x, r_x), y = (c_y, r_y), (c_x, c_y) \in \mathcal{C}^2, \textit{produits}(r_x) \cap \textit{substrats}(r_y) \neq \emptyset, r_x \neq r_y\}$$

avec les fonctions *produits*(r) et *substrats*(r) renvoyant respectivement l'ensemble des métabolites jouant le rôle de produits de r et l'ensemble des métabolites jouant le rôle de substrats de r.

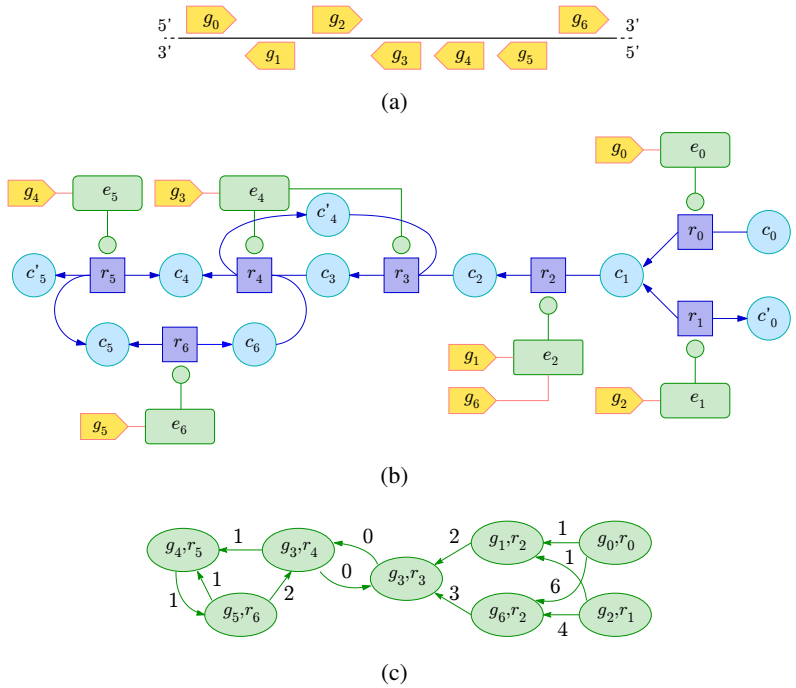


Figure 3.1 – Construction de $\mathcal{G}_{int}$ dans la version "gène" de SIPPER. Ici les unités catalytiques sont les gènes qui interviennent dans le réseau métabolique $\mathcal{M}$ (ce qui explique la notation g_i au lieu de c_i). Nous utilisons ici comme pondération la distance de colocalisation entre gènes (la différence entre les positions des gènes en nombre de gènes sur le génome). (a) Dans cet exemple, le génome est représenté comme un séquence linéaire de gènes (flèches épaisses), l'orientation de chaque gène indiquant son sens de lecture sur le génome. (b) Le réseau métabolique est fourni sous le standard SGBN : les métabolites (cercles) sont des substrats et/ou des produits de réactions (carrés), qui sont catalysées par des enzymes (rectangles arrondis) produites par des gènes (flèches épaisses). (c) Dans le modèle intégré résultant $\mathcal{G}_{int}$, chaque arc est pondéré par la distance qui existe entre les deux gènes de ses extrémités.

Poids w_d d'un arc Chaque arc $xy \in E$, avec $x = (c_x, r_x)$ et $y = (c_y, r_y)$, est pondéré par la valeur $w_d(x, y) = d(c_x, c_y)$. Le poids $w_d(x, y)$ qualifie la relation qui existe entre les sommets x et y. Plus la valeur de $w_d(x, y)$ est petite, meilleure est la qualité de la relation entre x et y.

Poids w_d d'un sous-graphe La qualité de cette relation peut-être étendue à n'importe quel sous-graphe de $\mathcal{G}_{int}$, que nous notons G'. Elle est alors le poids de G', noté $w_d(G')$, qui est la somme du poids des arcs de G'. Un exemple de construction de modèle intégré est visible dans la Figure 3.1.

$\mathcal{G}_{int}$ s'inspire ainsi d'un graphe orienté de réactions, dont les sommets sont couplés avec chacun des gènes ou des enzymes qui les catalysent. Le sommet correspondant à une réaction peut-être dupliqué (la réaction est associée à plusieurs gènes ou enzymes) ou disparaître (la réaction n'est pas catalysée) ; les arcs d'un sommet dupliqué étant dupliqués aussi. L'information métabolique est ainsi conservée, et l'information génomique ou enzymatique est ajoutée par (1) l'association des réactions avec des unités catalytiques, et (2) la pondération du graphe avec une distance dépendant des unités catalytiques. Nous

disposons donc de $\mathcal{G}_{int}$, à partir duquel nous cherchons à extraire des sous-graphes de façon automatique, chaque sous-graphe devant être biologiquement significatif. C'est ce que nous allons définir tout de suite.

3.3 La recherche automatique de sous-graphes biologiquement significatifs

Lors de l'analyse des réseaux métaboliques (Section 2.2.2), une technique souvent utilisée est la recherche de chemins, ou plus généralement, de sous-réseaux dans un réseau métabolique. Dans leurs travaux, Croes *et al.* [31] favorisent la recherche de plus courts chemins pour la découverte de voies métaboliques en utilisant une "distance métabolique". Les auteurs sélectionnent, parmi 5 plus courts chemins entre deux métabolites, celui qui correspond le plus à une voie métabolique déjà annotée. Bien qu'étant dans une démarche exploratoire d'un système bactérien sans autre connaissance que celle déjà intégrée, nous allons procéder de façon sensiblement similaire. Nous nous différencierons au niveau des critères de validation des chemins sélectionnés.

3.3.1 Notion de k-SIPs

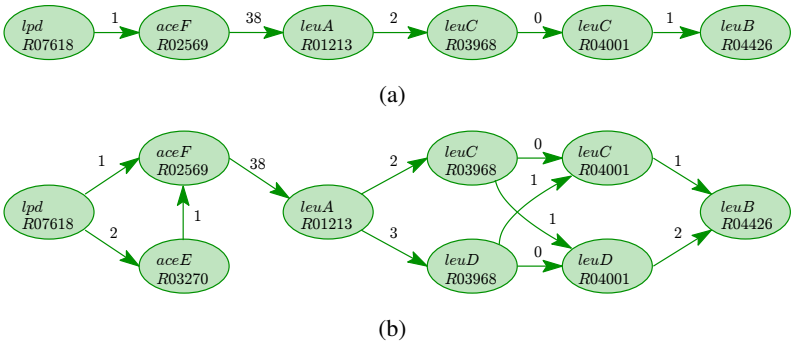
Un chemin dans $\mathcal{G}_{int}$ décrit un enchaînement de réactions transformant un métabolite en un autre, chaque réaction étant catalysée par une unité catalytique. Plusieurs chemins sont possibles dans $\mathcal{G}_{int}$, chacun ayant un poids w_d qualifiant l'enchaînement de réactions selon la distance d choisie. Comme pour un arc de $\mathcal{G}_{int}$, plus le poids w_d est petit, meilleure est la qualité de l'enchaînement. Il est donc naturel de chercher dans $\mathcal{G}_{int}$ les chemins qui minimisent la mesure w_d entre deux sommets x et y donnés.

Le fait qu'un chemin (i.e. un enchaînement de réactions) utilise deux fois une même réaction nous apparaît comme inintéressant. En effet, un chemin sur le réseau métabolique décrit la transformation d'un métabolite en un autre, et la présence d'un cycle indique que, outre l'augmentation du poids du chemin, une transformation déjà effectuée est refaite. Ce qui nous intéresse, c'est d'obtenir le plus directement un métabolite à partir d'un autre. Nous ajoutons donc la contrainte qu'un chemin ne doit pas passer plusieurs fois par une même réaction, c'est-à-dire qu'il doit être un chemin sans répétition de réactions.

Définition 8 (Chemin sans répétition de réactions ($srdr$-chemin))**.** *Soit $ch = s_1 s_2 \dots s_n$ un chemin de taille n dans $\mathcal{G}_{int}$. Le chemin ch est un chemin sans répétition de réaction (et par conséquent sans circuit), noté $srdr$-chemin, si $\forall i, j, 1 \leq i < j \leq n, s_i = (g_i, r_i), s_j = (g_j, r_j), r_i \neq r_j$.*

Pour deux ensembles de réactions donnés, nous aboutissons à la définition d'un ensemble de $srdr$-chemins, nommé k-SIP, qui respecte les contraintes précédentes :

Définition 9 (k-SIP)**.** *Soient R_1 et R_2 deux ensembles de réactions donnés tels que $R_1 \cap R_2 = \emptyset$, et k un entier donné. Un ensemble des k $srdr$-chemins de poids minimum qui vont des sommets contenant une réaction $r_1 \in R_1$ aux sommets contenant une réaction $r_2 \in R_2$ constitue un k-SIP (k shortest integrated paths) allant de R_1 à R_2 que nous notons k-SIP(R_1, R_2). Il n'y a pas unicité du k-SIP. En effet, il peut exister plusieurs plus courts $srdr$-chemins de même poids. Par abus de langage, lorsque $R_1 = \{r_1\}$ et $R_2 = \{r_2\}$ sont des singletons, nous utiliserons la notation k-SIP(r_1, r_2) plutôt que la notation k-SIP(R_1, R_2). De plus, nous utiliserons la notation k-SIP afin de désigner un k-SIP(R_1, R_2) quelconque, quelques soient R_1 et R_2.*



(a)

(b)

Figure 3.2 – Exemples de k-SIPs issus d'*E. coli* : (a) un 1-SIP ($\overline{w}_d = 7$) et (b) un 5-SIP ($\overline{w}_d = 7.43$) de la réaction $R07618$ à la réaction $R04426$.

Nous considérons qu'un k-SIP décrit un processus biologique potentiel. Il contient à la fois de l'information métabolique via les réactions que lui sont intégrées et de l'information génomique ou enzymatique selon la version de SIPPER utilisée. Quand $k = 1$ dans le cas de l'utilisation d'une métrique, le plus court chemin constituant le 1-SIP est un $srdr$-chemin. Lorsque $k > 1$, nous représentons les k-SIPs sous la forme d'un sous-graphe de $\mathcal{G}_{int}$ en superposant les chemins du k-SIP comme l'illustre la Figure 3.2.

3.3.2 Discrimination des k-SIPs intéressants

La recherche de k-SIP entre réactions permet de constituer un ensemble transformations métaboliques (i.e. les enchaînements de réactions) guidées par les relations qui existent en les unités catalytiques via le poids de chaque chemin. Afin de déterminer ceux qui sont les plus pertinents, nous devons les comparer entre eux. Le poids w_d de chaque chemin n'est pas forcément le critère le plus adapté pour comparer des k-SIPs entre eux. En suivant une démarche similaire à celle de Croes *et al.* [31], mais sans utiliser d'informations supplémentaires comme celle issues des annotations, nous proposons d'utiliser deux scores nommés le coefficient de voisinage $\overline{w}_d$ et la densité génomique d_G.

Coefficient de voisinage $\overline{w}_d$

Soit G' un sous-graphe de $\mathcal{G}_{int}$, $react(G')$ une fonction qui renvoie l'ensemble des réactions distinctes qui interviennent dans les couples (unité catalytique, réaction) qui constituent les sommets de G'. Le *coefficient de voisinage* $\overline{w}_d$ de G' est définit par la formule suivante :

$$\overline{w}_d(G') = \frac{w_d(G')}{|react(G')|}$$

avec $w_d(G')$ le poids de G'. Cette mesure est très proche d'un poids moyen des arcs du sous-graphe G', mais elle utilise les informations *omiques* intégrées dans $\mathcal{G}_{int}$ qui sont le poids de G', qui synthétise la distance d dans le sous-graphe, et le nombre de réactions qui participent à G'. Un sous-graphe G' a un petit $\overline{w}_d$ s'il inclut beaucoup de réactions, et si son poids $w_d(G')$ est faible. Inversement, il a un grand $\overline{w}_d$ s'il inclut peu de réactions et si son poids $w_d(G')$ est important. Dans notre cas, afin d'identifier les sous-graphes les plus intéressants, nous préférons ceux qui ont une petite valeur de $\overline{w}_d$. En effet, une

petite valeur de $w_d(G')$ indique une forte relation au niveau de constituants de G', normalisée par le nombre de réactions de G'.

Cette mesure $\overline{w}_d$ s'applique à un sous-graphe de $\mathcal{G}_{int}$, alors qu'un k-SIP est un ensemble de chemins. Lorsque $k > 1$ la mesure $\overline{w}_d(sip)$ d'un k-SIP nommé sip est la valeur de $\overline{w}_d(H)$ de H, le sous-graphe de $\mathcal{G}_{int}$ induit par les chemins composant sip.

Densité génomique d_G

Dans le cadre de l'intégration de l'information métabolique et génomique (version "gène" de SIPPER), nous avons défini une mesure, nommée la *densité génomique* d_G, pour discriminer un intervalle de gènes sur le génome qui participent ensemble à un même k-SIP nommé sip. Nous appelons ce type d'intervalle un intervalle induit par un ensemble de gènes :

Définition 10 (Intervalle minimum induit par un ensemble de gènes)**.** *Soit $\mathcal{G} = g_0 g_1 \dots g_n$ un génome unichromosomal, sous la forme d'un séquence linéaire ou circulaire de gènes et soit E un ensemble de gènes de $\mathcal{G}$. Un* intervalle minimum de gènes I_{min} induit par E dans $\mathcal{G}$ *est une sous-séquence minimale de $\mathcal{G}$ contenant les gènes de E. La notation $|I_{min}|$ indique la longueur de l'intervalle I_{min}, c'est-à-dire le nombre de gènes le constituant. Dans le cas des génomes linéaires, il y a toujours une seul intervalle minimum de gènes induit par un ensemble de gènes donné, mais dans le cas circulaire, il peut parfois y en avoir davantage.*

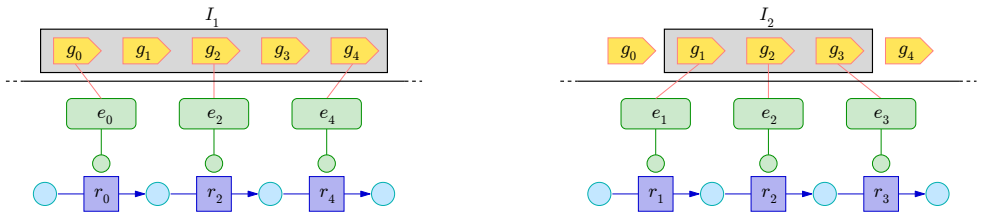
Soit $\textit{gènes}(sip)$ la fonction qui pour un k-SIP sip donné renvoie l'ensemble des gènes distincts qui participent à sip, et soit $\textit{intervalle}_{min}(\mathcal{G}, E)$ la fonction qui pour un ensemble de gènes donné du génome $\mathcal{G}$, retourne le plus petit intervalle de gènes induit par E dans $\mathcal{G}$. La *densité génomique* est définie par la formule suivante :

$$d_G(sip) = \frac{|\textit{gènes}(sip)|}{|\textit{intervalle}_{min}(\mathcal{G}, \textit{gènes}(sip))|}$$

La densité génomique prend des valeurs comprises entre 0 et 1. Soient sip un k-SIP donné et $I_{sip} = \textit{intervalle}_{min}(\mathcal{G}, \textit{gènes}(sip))$ l'intervalle minimum induit par les gènes présents dans sip. Lorsque $d_G(sip)$ tend vers 0, cela signifie que peu de gènes de I_{sip} participent à la catalyse de l'enchaînement de réactions de sip, tandis qu'une densité génomique de 1 indique que tous les gènes de I_{sip} participent à la catalyse de l'enchaînement de réactions de sip, c'est-à-dire que les gènes de sip sont successifs sur le génome (voir Figure 3.3). Cette mesure est donc utile pour mettre en évidence une relation de voisinage qui existe à la fois sur le génome (gènes successifs) et sur le réseau métabolique (réactions successives). Contrairement à $\overline{w}_d$, d_G ne dépend pas de la distance d utilisée dans $\mathcal{G}_{int}$, car d_G intègre déjà une distance entre gènes.

3.3.3 La recherche de k-SIPs

Un k-SIP, comme décrit dans la section 3.3.1, est un ensemble de k $srdr$-chemins de plus petit w_d entre deux ensembles de réactions donnés. La recherche des plus courts k $srdr$-chemins dans un graphe, même en se restreignant à $k = 1$, est un problème difficile à traiter. Dans un premier temps, nous allons décrire la recherche d'un plus court $srdr$-chemin dans un graphe orienté, et plus particulièrement dans une classe de graphe en particulier, appelée m-graphe orienté. Nous généraliserons ensuite à la recherche de k $srdr$-chemins. À chaque fois, nous ferons le parallèle entre $srdr$-chemin et k-SIPs. Enfin, à partir des résultats obtenus, nous nous intéresserons plus en détail à une variation des k-SIPs qui consiste à minimiser directement la mesure $\overline{w}_d$ plutôt que w_d.



(a) sip_1 (b) sip_2

Figure 3.3 – Illustrations de la densité génomique : (a) un enchaînement de réactions mettant en jeu un intervalle de gènes I_1 de densité $g_G(sip_1) = \frac{3}{5}$ et enchaînement de réactions mettant en jeu tous les gènes d'un intervalle I_2, ce qui donne une densité $d_G(sip_2) = 1$.

3.3.3.1 Problème du plus court $srdr$-chemin

Le problème de la recherche du plus court $srdr$-chemin consiste à rechercher le plus court chemin en interdisant la présence de certaines paires de sommets dans le chemin. Savoir si un tel chemin existe consiste, dans la littérature, à répondre au problème du chemin évitant les paires interdites (*paths avoiding forbidden pairs* en anglais ou **PAFP**) [62] qui est un problème NP-complet. Nous allons montrer, grâce à la façon dont est construit $\mathcal{G}_{int}$, qu'il existe toujours un chemin qui ne passe pas deux fois par une même réaction en travaillant sur une famille d'instances particulières du problème **PAFP** que nous nommons **PATP** (path avoiding twin pairs). Nous allons aussi montrer que lorsque le graphe est pondéré par une distance qui respecte l'inégalité triangulaire (i.e. une métrique), il est possible de proposer un chemin répondant au problème **PATP** en un temps polynomial.

Les réactions dédoublées de $\mathcal{G}_{int}$ forment des sommets jumeaux

Le problème **PAFP** est défini par Kolman et Pangrac [62] comme suit :

Définition 11 (**PAFP** [62])**.** *Étant donnés un graphe $G = (S, A)$, deux sommets fixés $s, t \in S$ et un ensemble F de paires interdites de sommets (appelées paires interdites), le problème du chemin évitant les paires interdites (**PAFP**) est de trouver un chemin de s à t qui contient au plus un sommet de chaque paire de F.*

Dans notre cas, nous cherchons un chemin qui ne passe pas plusieurs fois par une même réaction. Il existe deux cas. Le premier cas, le plus simple, concerne le cas où chaque réaction apparaît dans un seul sommet de $\mathcal{G}_{int}$. Dans ce cas là, il suffit de ne pas passer plusieurs fois par le même sommet, ce qui consiste à rechercher un chemin élémentaire, c'est-à-dire un chemin ne passant pas deux fois par un même sommet. Le second cas apparaît lorsqu'au moins une réaction est associée à plusieurs sommets de $\mathcal{G}_{int}$. Dans ce cas, par construction de $\mathcal{G}_{int}$, une propriété intéressante apparaît : la notion de sommets jumeaux. Sa définition utilise les notions de voisins entrant et sortant d'un sommet dans un graphe G, notés de la façon suivante :

Notation 1 (Voisins)**.** *Soit $G = (S, A)$ un graphe orienté et $x, y \in S$. Quel que soit $x \in S$, nous utiliserons les notations $N^+(x) = \{y \in S |\ xy \in A\}$ pour définir l'ensemble des voisins sortant de x dans G et $N^-(x) = \{y \in S |\ yx \in A\}$ pour définir l'ensemble des voisins entrant de x.*

Définition 12 (Jumeaux)**.** *Soit $G = (S, A)$ un graphe orienté sans boucle et $x, y \in S$. Le sommet x est un jumeau du sommet y si $N^+(x) = N^+(y)$, $N^-(x) = N^-(y)$. La fonction jumeaux(x) est alors définie*

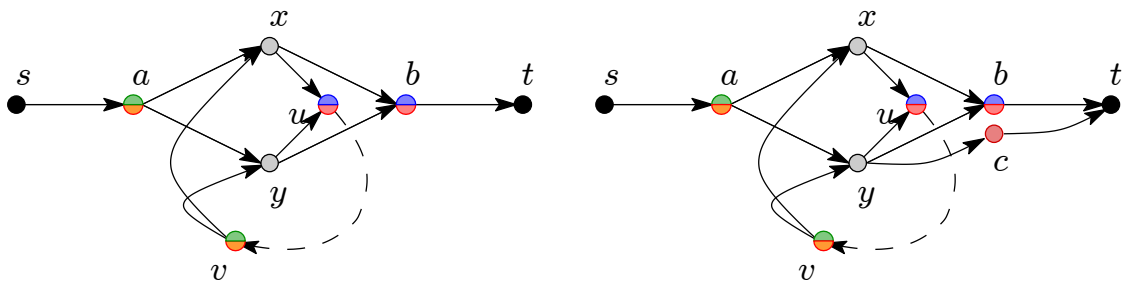


Figure 3.4 – Graphe orienté possédant ou non des jumeaux. Les sommets en verts indiques les sommets entrants de x (i.e. $N^-(x)$), ceux en orange les sommets entrants de y (i.e. $N^-(y)$), ceux en bleu les sommets sortants de x (i.e. $N^+(x)$) et ceux en rouge les sommets sortants de y (i.e. $N^+(y)$). Dans (a) les sommets x et y sont des jumeaux l'un de l'autre, tandis qu'ils ne le sont pas dans (b) car $N^+(x) \neq N^+(y)$.

comme la fonction qui associe à chaque $x \in S$ l'ensemble des jumeaux de x dans G (x non inclus). La Figure 3.4 illustre cette notion de sommets jumeaux.

D'après cette définition, lorsqu'une réaction r est catalysée par les unités catalytiques d'un ensemble $\mathcal{C}$, les sommets s_1 et s_2, avec $s_1, s_2 \in \{(c, r), c \in \mathcal{C}\}$, sont des jumeaux (i.e. $s_1 \in$ *jumeaux*(s_2) et $s_2 \in$ *jumeaux*(s_1)). Les sommets s_1 et s_2 sont *jumeaux* et constituent alors une *paire gémellaire*. Rechercher un chemin ch qui ne passe pas plusieurs fois par une même réaction dans $\mathcal{G}_{int}$ consiste donc à rechercher un chemin tel que, pour n'importe quelle paire de sommets de ch, cette paire de sommets ne soit pas gémellaire. Ce type de chemin est appelé *chemin sans jumeaux*.

Définition 13 (Chemin sans jumeaux)**.** *Soit $G = (S, A)$ un graphe orienté. Le chemin $ch = s_1 s_2 \dots s_n$ dans G est* sans jumeaux *si : pour tous $i, j \in \{1, \dots, n\}$ tel que $i \neq j$, nous avons $s_j \notin$ jumeaux(s_i).*

Dans G, la recherche d'un tel chemin entre deux sommets s et t de G forme le problème de la recherche de chemins sans jumeaux, nommé **PATP** (path avoiding twin pairs).

Définition 14 (PATP). *Étant donnés un graphe $G = (S, A)$, deux sommets fixés $s, t \in S$ et un ensemble $J = \{(x, y) |\ y \in$ jumeaux$(x)\}$ de paires gémellaires (*twin pairs *en anglais), le problème du chemin sans jumeaux, ou problème du chemin évitant les paires gémellaires (**PATP**) est de trouver un chemin de s à t qui contient au plus un sommet de chaque paire de J.*

Lorsqu'un chemin possède des jumeaux, il est possible d'obtenir un chemin sans jumeaux

Nous allons montrer que pour un chemin $ch = s \dots t$ avec jumeaux dans G et avec $s \notin$ *jumeaux*(t), il existe toujours un autre chemin entre s et t qui ne contient pas de jumeaux. Pour cela, nous avons besoin des notations qui suivent. Le chemin vide et un sous-chemin sont notés comme suit :

Notation 2 (Chemin vide)**.** *Le chemin vide, noté ϵ, désigne un chemin composé d'aucun sommet.*

Notation 3 (Longueur d'un chemin)**.** *Soit ch un chemin. La longueur de ch consiste au nombre d'arcs le constituant et est noté $||ch||$.*

Notation 4 (Sous-chemin)**.** *Soit $ch = s_1 s_2 \dots s_n$ un chemin dans un graphe G. Nous noterons $ch[s_i \dots s_j]$, avec $1 \leq i < j \leq n$, le sous-chemin de ch débutant par s_i et terminant par s_j.*

L'opération de concaténation de chemin, quant à elle, est notée de la façon suivante :

Notation 5 (Concaténation de chemins)**.** *Soient $ch_1 = s_1 s_2 \dots s_n$ et $ch_2 = t_1 t_2 \dots t_m$ deux chemins dans $G = (S, A)$. La concaténation de ch_1 et ch_2, notée par le symbole $\bullet$, consiste à créer le chemin $ch = ch_1 \bullet ch_2 = s_1 s_2 \dots s_n t_1 t_2 \dots t_m$. La concaténation est valide dans G si ch est un chemin de G. L'élément neutre de l'opération de concaténation de chemins est ϵ. Ainsi $ch \bullet \epsilon = \epsilon \bullet ch = ch$.*

Notation 6 (Concaténation d'un sommet et d'un chemin)**.** *Soient $ch = s_1 s_2 \dots s_n$ un chemin dans $G = (S, A)$, et $a \in S$. La concaténation de a à ch est possible si $s_n a \in A$ et s'écrit $ch' = ch \bullet a$. Le chemin ch' forme alors un chemin dans G. De même, la concaténation de ch à a est possible si $a s_1 \in A$ et s'écrit $ch'' = a \bullet ch$. Le chemin ch'' forme alors un chemin dans G.*

Notation 7 (Nombre de sommets jumeaux d'un chemin ($\#_j$))**.** *Soient $G = (S, A)$ un graphe orienté, s et $t \in S$ et ch un chemin entre s et t dans G. Le notation $\#_j(ch)$ indique le nombre de sommets de ch qui possède au moins un jumeau dans ch.*

Lemme 1. *Soient $G = (S, A)$ un graphe orienté, $s, t \in S$, $s \notin$ jumeaux(t) et ch un chemin entre s et t dans G. Si ch est un chemin avec jumeaux, alors il existe un chemin ch_0 entre s et t sans jumeaux, c'est-à-dire que $\#_j(ch_0) = 0$.*

Démonstration. Soit $G = (S, A)$ un graphe orienté et soit $ch = s \dots t$ un chemin tel que $s \notin$ *jumeaux*(t) et $i = \#_j(ch) > 0$. À partir de ch nous allons converger vers un chemin ch_0 tel que $\#_j(ch_0) = 0$.

Par définition de ch et sans perte de généralité, il existe deux sommets distincts x et y dans ch, tels que $y \in$ *jumeaux*(x). Nous avons donc $ch = s \dots axu \dots vyb \dots t$. D'après la Définition 12, nous avons dans G les propriétés suivantes : $a, v \in N^-(x) = N^-(y)$ et $b, u \in N^+(x) = N^+(y)$ (voir Figure 3.4 pour illustration). Par conséquent, il existe un chemin $ch' = ch[s \dots a] \bullet x \bullet ch[b \dots t]$ distinct de ch et avec $\#_j(ch') < \#_j(ch)$. En répétant ce procédé au plus i fois, nous obtenons un chemin ch_0 sans aucune paire gémellaire. Cela suffit à prouver que, quand il existe un chemin avec jumeaux entre s et t et que $s \notin$ *jumeaux*(t) dans G, il existe aussi un chemin sans jumeaux entre s et t. □

Ainsi, dans $\mathcal{G}_{int}$, s'il existe un chemin entre s et t avec s et t deux sommets sans jumeaux, alors soit ce chemin est sans jumeaux, soit il existe un autre chemin entre s et t qui est sans jumeaux.

Recherche d'un plus court chemin sans jumeaux

Dans notre cas, nous sommes surtout intéressés par la recherche de plus courts $srdr$-chemins qui sont des plus courts chemins sans jumeaux dans $\mathcal{G}_{int}$. Nous nommons ce problème le problème **SPATP** (shortest path avoiding twin pairs) défini comme suit :

Définition 15 (**SPATP**)**.** *Étant donnés un graphe $G = (S, A)$, deux sommets fixés $s, t \in S$ et un ensemble $J = \{(x, y) |\ y \in$ jumeaux$(x)\}$ de paires gémellaires, le problème du plus court chemin sans jumeaux (**SPATP**) est de trouver un plus court chemin de s à t qui contient au plus un sommet de chaque paire de J.*

Ainsi dans G un plus court chemin élémentaire (par rapport à w_d) entre deux sommets s et t est-il un chemin sans jumeaux et donc dans répétition de réactions ? Nous nous posons la question plus particulièrement quand G est un m-graphe (défini ci-dessous), c'est-à-dire que G est pondéré par une métrique,

distance qui vérifie l'inégalité triangulaire. Tout d'abord, nous allons montrer qu'il existe toujours un plus court chemin parmi l'ensemble des plus courts chemins qui est sans jumeaux quand G est pondéré par une métrique. Ensuite nous proposerons un algorithme exact qui calcule ce plus court chemin en un temps polynomial. Enfin, nous discuterons du cas où G est un graphe pondéré par une distance quelconque.

Notation 8 (m-graphe)**.** *Un m-graphe $G = (S, A)$ est un graphe pondéré par une métrique, c'est-à-dire que nous avons la propriété d'inégalité triangulaire au niveau du poids w_d de ses arcs : $\forall uv, vw, uw \in A, w_d(uv) + w_d(vw) \geq w_d(uw)$. Tout comme un graphe classique, un m-graphe peut-être orienté ou non-orienté.*

Plus court chemin sans jumeaux dans un m-graphe Le Lemme 1 nous indique que s'il existe au moins un chemin entre deux sommets non jumeaux dans un graphe, alors soit ce chemin est sans jumeaux, soit il en existe un autre sans jumeaux. De plus, par définition d'un m-graphe orienté $G = (S, A)$, nous avons la relation d'inégalité triangulaire au niveau du poids w_d de ses arcs. Ces deux propriétés nous font aboutir à une variation du Lemme 1 :

Lemme 2. *Soient $G = (S, A)$ un m-graphe orienté, $s, t \in S$, $s \notin$ jumeaux(t) et ch un chemin allant de s à t dans G. Si ch est un chemin avec jumeaux, alors il existe un autre chemin ch_0, sans jumeaux, allant de s à t tel que $w_d(ch_0) \leq w_d(ch)$ et $||ch_0|| < ||ch||$.*

Démonstration. Soit $G = (S, A)$ un graphe orienté et soit $ch = s \dots t$ un chemin tel que $s \notin$ *jumeaux*(t) et $i = \#_j(ch) > 0$. Par définition de ch et sans perte de généralité, il existe deux sommets x et y dans ch tel que $y \in$ *jumeaux*(x). Nous avons donc $ch = s \dots axu \dots vyb \dots t$. De plus, d'après la Définition 12, nous avons dans G les propriétés suivantes : $a, v \in N^-(x) = N^-(y)$ et $b, u \in N^+(x) = N^+(y)$ (voir Figure 3.4 pour illustration). Par conséquent, il existe un chemin $ch' = ch[s \dots a] \bullet x \bullet ch[b \dots t]$ distinct de ch avec $\#_j(ch') < \#_j(ch)$ possédant les propriétés suivantes :

- Par construction de ch' à partir de ch, nous avons $||ch'|| < ||ch||$.
- Selon le principe de l'inégalité triangulaire, nous avons :

$$w_d(xb) \leq w_d(xu) + w_d(ch[u \dots v]) + w_d(vy) + w_d(yb)$$

ce qui est équivalent à :

$$w_d(ch[s \dots a]) + w_d(ax) + w_d(xb) + w_d(ch[b \dots t])$$

$$\leq w_d(ch[s \dots a]) + w_d(ax) + w_d(xu) + w_d(ch[u \dots v]) + w_d(vy) + w_d(yb) + w_d(ch[b \dots t])$$

et donc $w_d(ch') \leq w_d(ch)$

De cette façon, nous montrons qu'il est possible de transformer un chemin ch possédant $\#_j(ch)$ sommets jumeaux vers un autre chemin ch' de poids moindre ou égal et possédant un nombre moindre de sommets jumeaux ($\#_j(ch') < \#_j(ch)$). En répétant ce processus au plus i fois, nous aboutissons sur un chemin ch_0 sans jumeaux. □

Du Lemme 2, nous remarquons la conséquence suivante :

Remarque 1. *Lorsqu'un chemin avec jumeaux ch est un plus court chemin selon la distance w_d, alors le chemin sans jumeaux ch_0 est aussi un plus court chemin. En effet, $w_d(ch_0) \leq w_d(ch)$ et $w_d(ch)$ est le poids minimum implique que $w_d(ch_0) = w_d(ch)$.*

Dans un graphe G, lorsqu'il existe plusieurs plus courts chemins entre s et t avec et sans jumeaux, comment est-il possible d'en discriminer un qui soit sans jumeaux ? Le théorème suivant nous indique une façon de faire.

Théorème 1. *Soit $G = (S, A)$ un m-graphe orienté et soit C l'ensemble non vide des plus courts chemins entre s et t dans G. Le plus court chemin $ch \in C$ qui possède le nombre minimum d'arcs parmi tous les chemins de C est un chemin sans jumeaux.*

Démonstration. Procédons par l'absurde. Soit ch un chemin de C tel que $\#_j(ch) > 0$ et $||ch|| = \min\{||p||, \forall p \in C\}$. Le Lemme 2 nous indique qu'il existe un chemin ch' distinct de ch tel que $w_d(ch') \leq w_d(ch)$ et $||ch'|| < ||ch||$. Alors $w_d(ch') = w_d(ch)$ donc $ch' \in C$ et $||ch'|| < ||ch||$, ce qui entre en contradiction avec la proposition initiale. Par conséquent, le Théorème 1 est valide. □

Calcul du plus court chemin sans jumeaux dans un m-graphe D'après le Théorème 1, le chemin de poids minimum avec le moins d'arcs est sans jumeaux. Cela correspond à la recherche d'un plus court chemin multiobjectif [41] dans $G = (S, A)$ avec un ordre lexicographique sur les critères à optimiser qui sont ici la minimisation du poids du chemin puis la minimisation de sa longueur (i.e. le nombre d'arcs). Pour obtenir un tel chemin, nous avons besoin de définir les notions et notations qui suivent.

Notation 9 (coût d'un arc)**.** *À chaque arc $ab \in A$ sont associés q coûts, qui sont contenus dans un vecteur de coûts $f(ab)$. La notation $f_i(ab)$ permet d'accéder à la i-ième valeur du vecteur de coûts $f(ab)$ avec $1 \leq i \leq q$. Nous dirons qu'un vecteur de coût $f(ab)$ est positif si pour tout $1 \leq i \leq q$ la propriété $f_i(ab) \geq 0$ est vérifiée.*

Notation 10 (vecteur de coûts nul : $\overrightarrow{0}$)**.** *Un vecteur de coût x est un vecteur de coût nul si $x = \overrightarrow{0}$ = (0,..., 0).*

Notation 11 (vecteur de coûts infini : $+\overrightarrow{\infty}$)**.** *Un vecteur de coût x est un vecteur de coût infini si $x = +\overrightarrow{\infty} = (+\infty, \ldots, +\infty)$*

Définition 16 (addition de vecteurs de coûts)**.** *Soient $x = (x_1, x_2, \ldots, x_q)$ et $y = (y_1, y_2, \ldots, y_q)$ deux vecteurs de coûts de taille q. L'addition $(+)$ de x et y se réalise de la façon suivante : $x + y = (x_1 + y_1, x_2 + y_2, \ldots, x_q + y_q)$*

Définition 17 (coût d'un chemin)**.** *Soit $ch = s_1 s_2 \ldots s_n$ un chemin de taille n. Le coût f du chemin ch est la somme du coût de ses arcs, c'est-à-dire $f(ch) = \sum_{i=1}^{n-1} f(s_i s_{i+1})$. La notation $f_i(ch)$ permet d'accéder à la i-ième valeur du vecteur de coûts $f(ch)$ avec $1 \leq i \leq q$.*

Définition 18 (ordre lexicographique entre deux vecteurs de coûts)**.** *Soient $x = (x_1, x_2, \ldots, x_q)$ et $y = (y_1, y_2, \ldots, y_q)$ deux vecteurs de coûts de taille q. L'ordre lexicographique $\prec_{lex}$ entre ces deux coûts est défini comme suit :*

$$x \prec_{lex} y \text{ si et seulement si : } \begin{cases} & x_1 < y_1 \\ \text{ou} & x_1 = y_1 \text{ et } x' = (x_2, \ldots, x_q) \prec_{lex} y' = (y_2, \ldots, y_q) \end{cases}$$

Cette relation d'ordre est importante, car il est possible comparer plusieurs vecteurs de coûts entre-eux ainsi que d'utiliser des fonctions de classement des vecteurs de coûts comme les fonctions min *et* max.

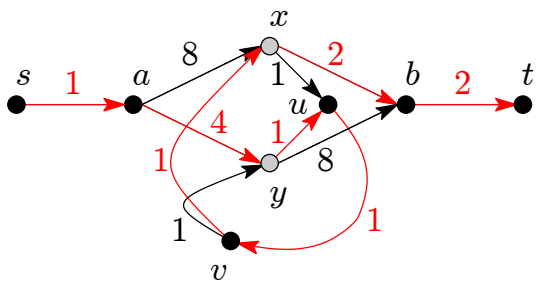


Figure 3.5 – Exemple d'un plus court chemin passant par deux sommets jumeaux. Les sommets x et y sont jumeaux. Le plus court chemin allant de s à t est indiqué en rouge et passe par x et y.

Lorsqu'un ordre lexicographique sur les critères est défini, la recherche d'un plus court chemin multiobjectif à partir de s se réalise facilement à l'aide de l'algorithme de `Dijkstra` [34] (Algorithme 4) L'algorithme de `Dijkstra` est un algorithme glouton qui calcule le coût minimum pour atteindre un sommet t à partir d'un initial s. Pour cela il réalise, pour chaque sommet $v \in S$, des approximations successives du coût d'accès minimum à partir de s noté $f'(v)$ (voir Algorithme 2). Le cheminement suivi au travers de G pour accéder à v à partir de s est mémorisé en conservant le prédécesseur de v, noté $pred(v)$ et qui est l'avant-dernier sommet d'un chemin de s à v ayant pour coût $f'(v)$. Lorsqu'un sommet n'a pas de prédécesseur, $pred(v)$ à pour valeur `nil`. Il est ainsi possible de reconstruire le chemin, lorsqu'il existe, entre s et t prenant successivement les prédécesseurs de t jusqu'à s (voir Algorithme 3). Au début de l'algorithme, chaque sommet $v \in V$ est initialisé de la sorte : $f'(v)$ est infini, sauf $f'(s)$ qui est nul, et $pred(v)$ vaut `nil` (voir Algorithme 1).

Dans le cadre de la résolution de **SPATP**, chaque arc est pondéré par w_d. Afin d'obtenir des plus courts chemins sans jumeaux, le vecteur de coûts d'un arc $ab \in A$ est : $f(ab) = (w_d(ab), 1)$. Le coût $f_1(ab)$ est le poids de l'arc ab et $f_2(ab)$ est la longueur de ab. Ainsi, pour un chemin ch, $f_1(ch)$ est le poids de ch et $f_2(ch)$ indique sa longueur. La relation d'ordre utilisée est alors la relation d'ordre lexicograpique $\prec_{lex}$ (voir Définition 18).

Calcul du plus court chemin sans jumeaux dans graphe quelconque Dans le cas d'un graphe G pondéré par une distance quelconque, il est difficile de proposer un algorithme exact qui calcule, entre deux sommets s et t, un chemin sans jumeaux le plus petit possible. Bien que le Lemme 1 nous garantisse l'existence d'un chemin sans jumeaux, découvrir le plus court possible n'est pas aussi simple que dans le cas d'un m-graphe où l'inégalité triangulaire nous garantit qu'il y a toujours, parmi l'ensemble des plus courts chemins, un chemin sans jumeaux. La Figure 3.5 le montre bien : le plus court chemin (en rouge) passe par x et y qui sont jumeaux l'un de l'autre et il n'existe pas de chemin de poids identique sans jumeaux. Une solution, non exacte mais simple, consiste à calculer un plus court chemin $ch = s_1 s_2 \dots s_n$. Sans perte de généralité, si ch possède des paires de jumeaux (s_i, s_j), nous remplaçons le sous-chemins $ch[s_i \dots s_j]$ avec $j \in \textit{jumeaux}(i)$ par s_i tel que $w_d(s_{i-1}s_i s_{j+1}) \leq w_d(s_{i-1}s_j s_{j+1})$. Nous n'avons pas la garantie d'avoir le plus court chemin sans jumeaux, mais le chemin que nous aurons sera au moins sans jumeaux et assez court.

Algorithme 1 `INIT` : Initialisation de l'algorithme de Dijkstra

1: **Pour tout** $v \in S$ **faire**
2: $f'(v) \leftarrow +\overrightarrow{\infty}$
3: *pred*$(v) \leftarrow$ `nil`
4: **Fin pour**
5: $f'(s) \leftarrow \overrightarrow{0}$

Algorithme 2 `RELAX`(v, a) : Relaxation de l'arc va

1: **Si** $f'(v) + f(va) \prec f'(a)$ **alors**
2: $f'(a) \leftarrow f'(v) + f(va)$
3: *pred*$(a) \leftarrow v$
4: **Finsi**

Algorithme 3 `ConstructPath` : Construit le plus court chemin de s à t

1: $ch \leftarrow t$
2: $v \leftarrow t$
3: **Tantque** $v \neq s$ **faire**
4: $v \leftarrow$ *pred*(v)
5: $ch \leftarrow v \bullet ch$
6: **Fin tantque**

Algorithme 4 Algorithme de `Dijkstra` : Algorithme de recherche du plus court chemin.

ENTRÉES: $G = (S, A)$ un graphe orienté et pondéré,
$s \in S$ le sommet initial du plus court chemin,
$t \in S$ le sommet final du plus court chemin et
$\prec$ une relation d'ordre entre deux vecteurs de coûts.
SORTIES: un plus court chemin de s à t dans G.

1: `INIT`
2: $Q \leftarrow S$
3: **Tantque** $Q \neq \emptyset$ **faire**
4: Soit z tel que $f'(z) = \min\{f'(v), v \in Q\}$
5: Retirer z de Q
6: **Pour tout** $a \in N^+(z)$ **faire**
7: `RELAX`(z, a)
8: **Fin pour**
9: **Fin tantque**
10: `ConstructPath`
11: **Retourner** ch

3.3.3.2 Problème des k plus courts $srdr$-chemins

Lorsque $\mathcal{G}_{int}$ est pondéré par une métrique (c'est-à-dire avec inégalité triangulaire), nous sommes capables de calculer un plus court chemin sans jumeaux entre deux réactions données (donc sans répétition de réactions). Cependant ce n'est pas seulement le plus court chemin qui nous intéresse pour le calcul des k-SIPs, mais les k-plus courts chemins entre deux ensembles de réactions distinctes (i.e. de sommets distincts). De quelle façon pouvons nous alors procéder ? Nous allons, dans un premier temps, présenter l'algorithme de Yen qui permet de calculer, dans un graphe orienté quelconque G, les k plus courts chemins élémentaires entre deux sommets, puis nous ajouterons un prétraitement de G permettant de calculer les k plus courts chemins entre deux ensembles de sommets et enfin, nous proposerons une modification de l'algorithme de Yen qui permet le calcul des k plus courts chemins sans jumeaux entre deux sommets dans un m-graphe.

Calcul des k plus courts chemins : l'algorithme de Yen

Afin de calculer, dans un graphe orienté $G = (S, A)$, l'ensemble des k plus courts chemins entre s et t, Yen [108] a proposé un algorithme exact qui calcule d'abord le plus court chemin, puis le second plus court, et ainsi de suite jusqu'au k-ième plus court chemin. La complexité temporelle de cet algorithme est de l'ordre de $O(kn(m + n \log n))$ avec k le nombre de chemins recherchés, n le nombre de sommets du graphe G et m le nombre d'arcs de G.

Dans un graphe G, étant donné C_p l'ensemble des $p < k$ plus courts chemins entre s et t, la recherche du $(p + 1)$-ième plus court chemin, distinct de ceux de C_p, s'effectue en perturbant le graphe G. Cela consiste au retrait d'un ou plusieurs arcs des chemins de C_p, et de rechercher le nouveau un plus court chemin dans G entre s et t. L'algorithme de Yen utilise les notions de préfixe et de point de déviation entre chemins pour parvenir à ce résultat. La notion de *préfixe commun* indique à quel point deux ou plusieurs chemins débutent de façon identique.

Définition 19 (Préfixe d'un chemin)**.** *Soit $ch = s_1 s_2 \cdots s_n$ un chemin élémentaire, c'est-à-dire un chemin ne passant pas plusieurs fois par un même sommet, de longueur $n - 1$ dans un graphe G. Un préfixe p de ch est un sous-chemin $p = s_1 s_2 \cdots s_m$ de ch avec $m < n$. La fonction $prefixe(ch, m)$ permet d'obtenir ce préfixe p du chemin ch de longueur $m - 1$.*

Définition 20 (Préfixe commun à deux chemins)**.** *Soient ch_1 et ch_2 deux chemins dans G. Un préfixe commun p à ch_1 et ch_2 est un chemin de G qui est à la fois préfixe de ch_1 et de ch_2. Il se peut que ch_1 et ch_2 n'admettent aucun préfixe commun, nous dirons alors que le préfixe commun à ch_1 et ch_2 est ϵ. Le préfixe commun maximum p_{max} entre ch_1 et ch_2 est le plus grand préfixe commun à ch_1 et ch_2.*

Il est important de remarquer que, dans le cadre de la recherche de k plus courts chemins entre s et t, tous les chemins auront pour préfixe commun le sommet s. Cela nous apporte la garantie qu'il existe toujours un préfixe commun entre deux chemins distincts allant de s à t, et aussi qu'il existe toujours un sommet à partir duquel les deux chemins diffèrent. Ce sommet se nomme le *point de déviation* entre les deux chemins :

Définition 21 (Point de déviation entre deux chemins)**.** *Soient $ch_1 = s_1 s_2 \cdots s_i s_j \cdots s_n$ et $ch_2 = s_1 s_2 \cdots s_i s_k \cdots s_m$ deux chemins dans G tel que $s_j \neq s_k$, et soit $p_{max} = s_1 s_2 \cdots s_i \neq \epsilon$ le préfixe commun maximum à ch_1 et ch_2. Le* point de déviation *entre ch_1 et ch_2 est s_i, le dernier sommet du préfixe commun maximum à ch_1 et ch_2.*

L'arbre des préfixes communs maximaux est une structure de données définie à l'aide des notions de préfixe et point de déviation. C'est une structure de données qui permet de stocker de façon compacte un ensemble de chemins et qui n'est pas sans rappeler la structure d'arbre des suffixes, largement utilisée dans le cadre de l'indexation de texte [73]. Soit C un ensemble de chemins tel que, pour tous chemins $ch_1, ch_2 \in C$, ch_1 et ch_2 ont un préfixe commun, c'est-à-dire qu'ils ont au moins leur premier sommet, nommé s, en commun. L'arbre des préfixes communs maximaux T de C est construit comme suit. La racine de T est s, et chaque parcours entre la racine et une feuille de T correspond à un chemin de C. Deux chemins distincts de C suivent le même parcours dans T tout au long de leur préfixe commun maximum, et se séparent pour aller dans deux sous-arbres différents dans A au niveau de leur point de déviation. Ainsi, chaque nœud de degré supérieur à 1 dans T est un point de déviation entre les chemins passant de part et d'autre de ce nœud. Il devient alors facile de connaître tous les préfixes communs maximaux et points de déviation d'un chemin ch stocké dans T avec n'importe quel autre chemin de T. Lorsque T représente un seul chemin ch, le point de déviation de ch dans T est par convention la racine de T. Pour connaître le nombre de chemins stockés dans T, il suffit de connaître le nombre de feuilles de T, car chaque feuille est le sommet de terminaison d'un chemin. Nous notons $largeur(T)$ le nombre de feuilles de T. La Figure 3.6 est un exemple d'arbre des préfixes.

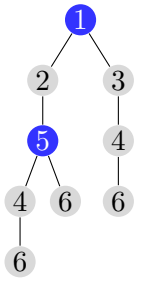


Figure 3.6 – Arbre des préfixes communs maximaux des trois chemins 12546, 1256 et 1346. Les nœuds en bleu représentent les points de déviation entre les chemins se séparant au niveau de ces nœuds (nœuds 1 et 5) et l'arbre est de largeur 3.

L'algorithme de `Yen` repose sur la propriété que pour un plus court chemin $ch = s_1 s_2 \dots s_n$ entre s_1 et s_n dans $G = (S, A)$, $\forall i, j$ avec $1 \leq i < j \leq n$, le sous-chemin de ch entre s_i et s_j est aussi un plus court chemin entre s_i et s_j dans G. Cela signifie qu'un plus court chemin est composé de plus courts chemins. Ainsi, dans G la recherche d'un nouveau plus court chemin allant de s à t consiste en la recherche de plus courts sous-chemins distincts de ceux déjà connus. Cet algorithme utilise l'algorithme de `Dijkstra` (Algorithme 4), où le coût d'un arc $ab \in A$ est son poids $w_d(ab)$ et la relation d'ordre utilisée est la relation d'infériorité $<$. Suite à la recherche du plus court chemin initial (ligne 2), de nouveaux plus courts chemins sont étudiés. Pour cela, le graphe G est perturbé par le retrait de sommets et d'arcs ligne 8. L'arbre des préfixes communs maximaux, utilisé à la ligne 6, permet de connaître quels sont les plus courts chemins qui ont déjà été calculés grâce aux notions de préfixe commun maximum et point de déviation. Il permet donc de ne pas recalculer des chemins déjà présents dans les *Chemins candidats*, seuls de nouveaux sous-chemins entre le point de déviation et la fin du *chemin étudié* sont calculés car ce sont les seuls sous-chemins qui manquent pour choisir le prochain plus court chemin (lignes 9 à 17). Ces opérations, de la ligne 3 à la ligne 19, sont répétées p fois avec $p \leq k$, générant ainsi p plus courts chemins, étant donné qu'il n'existe a pas toujours k chemins distincts entre deux sommets dans un graphe. Un exemple d'exécution de l'algorithme est disponible dans l'Annexe A.

Algorithme 5 Algorithme de `Yen` : Algorithme de recherche des k plus courts chemins sans circuit.

ENTRÉES: $G = (S, A)$ un graphe orienté, k le nombre de plus courts chemins désirés, $s \in V$ le sommet initial des k plus courts chemins, $t \in V$ le sommet final des k plus courts chemins.

SORTIES: une liste d'au plus k plus courts chemins de s à t dans G.

Soit T un arbre vide.
Soit *Chemins candidats* un ensemble singleton constitué du résultat de `Dijkstra`$(G, s, t, <)$
Tantque *Chemins candidats*$\neq \emptyset$ **et** $largeur(T) < k$ **faire**
    Soit *chemin étudié*$= s_1 \dots s_n$ un chemin de plus petit poids dans *Chemins candidats*, avec $s_1 = s$ et $s_n = t$.
    Retirer *chemin étudié* de *Chemins candidats* et l'ajouter dans T.
    Dans T, soit s_d le point de déviation le plus profond de *chemin étudié*, et toujours dans T, soit F l'ensemble des fils de s_d.
    Soit $p =$*chemin étudié*$[s_1 \dots s_{d-1}]$.
    Dans G, retirer tous les sommets de p et tous les arcs $s_d b$ avec $b \in F \backslash \{s_{d+1}\}$.
    Pour $i \leftarrow d$ **à** $n - 1$ **faire**
        Dans G, retirer l'arc $s_i s_{i+1}$ issu de *chemin étudié*.
        Soit ch le résultat de `Dijkstra`$(G, s_i, t, <)$.
        Si ch existe **alors**
            Ajouter $p \bullet ch$ à la liste des *Chemins candidats*.
        Finsi
        $p \leftarrow p \bullet s_i$.
        Dans G retirer s_i.
    Fin pour
    Remettre dans G les arcs et les sommets supprimés.
Fin tantque
Retourner les p chemins les plus courts ($p \leq k$), stockés dans T, sous la forme d'une liste de chemins.

Algorithme 6 L'algorithme `ComputeSIPs`

ENTRÉES: $\mathcal{G}_{int}$ le modèle intégré,
R_1 un ensemble de réactions initiales,
R_2 un ensemble de réactions finales tel que $R_1 \cap R_2 = \emptyset$ et
k le nombre de plus courts chemins désirés.
SORTIES: un k-SIP entre R_1 et R_2 dans $\mathcal{G}_{int}$.
1: Soit H une copie de $\mathcal{G}_{int}$
2: Dans H, contracter tous les sommets (c_1, r_1) avec $r_1 \in R_1$ dans un unique sommet (C_1, R_1)
3: Dans le H résultant, contracter tous les sommets (c_2, r_2) avec $r_2 \in R_2$ dans un unique sommet (C_2, R_2)
4: Calculer $sip \leftarrow \texttt{Yen}(H, k, (C_1, R_1), (C_2, R_2))$
5: Soit k-SIP(R_1, R_2) l'ensemble des k chemins induits dans $\mathcal{G}_{int}$ par sip

Prétraitement pour rechercher les k plus courts chemins entre ensembles de réactions

L'algorithme de `Yen`, dans un graphe orienté G pondéré et pour un entier k donné, permet la recherche des k plus courts chemins allant d'un sommet s à un sommet t. Mais par définition, un k-SIP de $\mathcal{G}_{int}= (V, E)$ est l'ensemble des plus courts chemins allant d'un ensemble de réactions nommé R_1 à un autre nommé R_2, c'est-à-dire d'un ensemble de sommets $D = \{(g, r) : r \in R_1\}$ à un ensemble de sommets $F = \{(g, r) : r \in R_2\}$. L'algorithme `ComputeSIPs` (Algorithme 6) nous permet de réaliser le calcul de chemins entre ces deux ensembles en utilisant l'astuce suivante : dans $\mathcal{G}_{int}$, en contractant les sommets de D en un nouveau sommet s et en contractant ceux de F en un sommet t, le calcul des k plus courts chemins allant de s à t revient à calculer les k plus courts chemins allant de R_1 à R_2.

Cependant, ce traitement ne nous permet pas d'éviter les répétitions de réactions dans chaque chemin composant un k-SIP. Il nous faut donc modifier l'algorithme de `Yen` pour obtenir ce type de chemins.

Modification de l'algorithme de `Yen` pour la prise en compte des particularités de $\mathcal{G}_{int}$.

L'algorithme de `Yen` recherche les k plus courts chemins sans circuit, c'est-à-dire que chaque chemin ne passe plusieurs fois par le même sommet du graphe. Dans notre cas, nous souhaitons non seulement que chaque chemin ne passe pas plusieurs fois par le même sommet ; mais nous souhaitons également qu'il ne passe pas par des sommets jumeaux. Nous nommons le problème recherche de k plus courts chemins sans jumeaux le problème **KSPATP** (k shortest paths avoiding twin pairs). Nous nous sommes attachés en particulier à résoudre ce problème dans le cas des m-graphes.

Dans le cas d'un m-graphe orienté, seules quelques modifications sont à apporter à l'algorithme de `Yen` (Algorithme 5) pour la recherche des k plus courts chemins sans jumeaux, pour un k donné. L'algorithme `ComputeKSPATP` (voir Algorithme 7) est l'algorithme modifié en conséquence. Tout d'abord, la recherche du plus court chemin, aux lignes 2 et 11, consiste à résoudre une instance du problème **SPATP**. Pour chaque arc $ab \in A$, son vecteur de coûts est $f(ab) = (w_d(ab), 1)$, et la relation d'ordre utilisée est la relation d'ordre lexicograpique $\prec_{lex}$ (voir Définition 18). Puis, lors de la perturbation du graphe G qui permet le calcul de nouveaux sous-chemins alternatifs, nous retirons aux lignes 8 et 10, en plus des sommets pris en compte dans le préfixe p, les jumeaux des sommets de p. Ainsi, la concaténation du préfixe et du nouveau sous-chemin calculé génère un nouveau plus court chemin candidat sans jumeaux.

Algorithme 7 Algorithme `ComputeKSPATP` : Algorithme de recherche des k plus courts chemins sans jumeaux.

ENTRÉES: $G = (V, E)$ un m-graphe orienté,
k le nombre de plus courts chemins désirés,
$s \in V$ le sommet initial des k plus courts chemins et
$t \in V$ le sommet final des k plus courts chemins.
SORTIES: une liste de au plus k plus court chemins sans jumeaux de s à t dans G.

Soit T un arbre vide.
Soit *Chemins candidats* un ensemble singleton constitué du résultat de $Dijkstra(G, s, t, \prec_{lex})$
Tantque *Chemins candidats*$\neq \emptyset$ **et** $largeur(A) < k$ **faire**
  Soit *chemin étudié* $= s_1 \dots s_n$ le chemin de plus petit poids dans *Chemins candidats*, avec $s_1 = s$ et $s_n = t$.
  Retirer *chemin étudié* de *Chemins candidats* et l'ajouter dans A.
  Dans T, soit s_d le point de déviation le plus profond de *chemin étudié*, et toujours dans T, soit F l'ensemble des fils de s_d.
  Soit $p =$ *chemin étudié*$[s_1 \dots s_{d-1}]$.
  Dans G, retirer tous les sommets de p, *ceux de jumeaux*(s_i) *avec* $1 \leq i \leq d-1$ et tous les arcs $s_d b$ avec $b \in F \backslash \{s_{d+1}\}$.
  Pour $i \leftarrow d$ **à** $n-1$ **faire**
    Dans G, retirer l'arc $s_i s_{i+1}$ issu de *chemin étudié ainsi que les sommets de jumeaux*(s_i).
    Soit ch le résultat de $Dijkstra(G, s_i, t, \prec_{lex})$.
    Si ch existe **alors**
      Ajouter $p \bullet ch$ à la liste des *Chemins candidats*.
    Finsi
    $p \leftarrow p \bullet s_i$.
    Dans G retirer s_i.
  Fin pour
  Remettre dans G les arcs et les sommets supprimés.
Fin tantque
Retourner les k chemins les plus courts, stockés dans T, sous la forme d'une liste de chemins.

Ces simples modifications, indiquées en bleu, suffissent à obtenir les k plus courts chemins sans circuit ni jumeaux dans un m-graphe.

Dans le cas d'un graphe orienté quelconque, tant que nous n'avons pas de solution exacte pour la recherche du plus court chemin sans jumeaux, il nous est difficile, pour un entier k donné, de proposer une solution exacte pour la recherche des k plus courts chemins sans jumeaux, et donc sans répétition de réaction. Dans ce cas là nous irons au plus simple : nous utiliserons l'algorithme de `Yen` pour calculer des plus courts chemins.

3.3.4 Recherche de $srdr$-chemins minimisant $\overline{w}_d$

Nous avons défini un k-SIP comme un ensemble de k plus courts $srdr$-chemins, c'est-à-dire un ensemble de k plus courts chemins sans jumeaux ni circuit. Nous avons également indiqué nous intéresser au coefficient de voisinage $\overline{w}_d$ de chaque k-SIP. Étant donnée la définition de $\overline{w}_d$ (voir page 28), les k plus

courts chemins sans répétition de réaction dans un m-graphe ne sont pas forcément ceux de plus petit $\overline{w}_d$. En effet, dans les algorithmes `ComputeKSPATP` et `Dijkstra`, lorsqu'il existe plusieurs possibilités pour un plus court chemin, nous choisissons celui qui possède le moins d'arcs, et donc le moins de réactions. Ce chemin ch possède ainsi un $\overline{w}_d$ plus grand que les autres plus courts chemins sans jumeaux possibles, le poids de chacun des chemins étant le même que le poids de ch. Dans le but de chercher une variante d'un k-SIP qui minimise $\overline{w}_d$, l'utilisation de l'algorithme `ComputeKSPATP` ne fourni pas de résultat exact. Tout d'abord, nous allons étudier la complexité de la recherche du $srdr$-chemin minimisant $\overline{w}_d$ et montrer qu'intuitivement ce problème semble difficile, puis nous allons présenter une méthode non exacte pour calculer de tels chemins.

Étude de la complexité

Dans $\mathcal{G}_{int}$, pour R_1 et R_2 deux ensembles de réactions donnés, la recherche d'un $srdr$-chemin entre R_1 et R_2 qui minimise $\overline{w}_d$ est difficile. Nous allons présenter cette difficulté en travaillant sur une version restreinte de $\mathcal{G}_{int}$, notée $\mathcal{H}$, où chaque réaction apparaît dans un unique sommet. Dans ce cas, un $srdr$-chemin dans $\mathcal{H}$ est un chemin sans circuit, et la valeur de $\overline{w}_d$ est alors égale au ratio du poids du chemin et nombre de sommets intervenant dans le chemin, soit la longueur du chemin plus un. Plusieurs problèmes de la littérature semblent intuitivement proches de notre problème du $srdr$-chemin minimisant $\overline{w}_d$ lorsque nous nous intéressons à $\mathcal{H}$. Le premier est celui de la recherche du chemin minimisant le poids moyen de ses arcs dans un graphe :

Définition 22 (Le problème de la recherche du chemin minimisant le poids moyen de ses arcs)**.** *Soient $G = (S, A)$ un graphe orienté dont les arcs sont pondérés, et P un chemin entre s et t dans G. Le poids de P est la somme du poids de ses arcs, et le poids moyen des arcs de P, noté $w_{moy}(P)$, est le ratio du poids de P et de la longueur de P (i.e le nombre d'arcs de P). Le problème de la recherche du chemin minimisant le poids moyen de ses arcs consiste à trouver un chemin P entre s et t minimisant $w_{moy}(P)$.*

Lorsque le graphe étudié est acyclique, ce problème se résout en un temps polynomial [104]. Cependant, les graphes biologiques que nous étudions contiennent des circuits. De plus, nous travaillons avec la mesure $\overline{w}_d$ qui est le ratio du poids du chemin et du nombre de sommets du chemin, et non du nombre d'arcs.

Le second problème proche est celui du *all-pairs minimum average weighted length path* qui est définit comme suit :

Définition 23 (Le problème *all-pairs minimum average weighted length path*)**.** *Soit $G = (S, A)$ un graphe orienté dans lequel chaque arc $ij \in A$ possède deux poids, représentant une longueur et un temps de trajet. La longueur de n'importe quel chemin P dans G est définie comme la somme de la longueur des arcs de P. Le temps de trajet de P est défini de manière analogue comme étant la somme des temps de trajet des arcs de P. Le problème* all-pairs minimum average weighted length path *consiste à trouver un chemin P entre chaque paire de nœuds dans S tel que le ratio de la longueur de P et de temps du trajet de P soit minimum.*

Ce problème est NP-Complet [106], cependant il existe un algorithme pseudo-polynomial le résolvant lorsque le graphe G ne contient pas de *tadpole*, une structure analogue à un circuit négatif dans un plus court chemin. De plus, lorsque les temps de trajets sont égaux (mais pas les longueurs) dans ce graphe sans tadpole, le problème se résout en un temps polynomial. En donnant un temps de trajet de 1 à chaque arc, nous pourrions utiliser ce problème pour approcher le nôtre. Cependant, les solutions du problème all-pairs minimum average weighted length path peuvent contenir des circuits positifs, pour

peu que le circuit fasse diminuer le score du chemin. L'utilisation de ce problème pour résoudre la recherche de $srdr$-chemin implique que nous acceptions la présence de circuit dans notre $srdr$-chemin, qui par définition, n'admet pas de circuit.

Si nous fixons une longueur de chemins l avec $1 \leq l \leq |V|$, trouver le chemin de longueur l minimisant $\overline{w}_d$ consiste à trouver un chemin de longueur l minimisant w_d. Ce problème est très proche de celui de la recherche du plus court chemin restreint (restricted shortest path problem) [47, 81] défini comme suit :

Définition 24 (Problème du plus court chemin restreint)**.** *Soient $G = (V, E)$ un graphe orienté, s et t deux sommets de G et L un entier. Chaque arc xy de G est pondéré par deux mesures nommées $l(xy)$ et $w(xy)$. Dans G, le poids du chemin $ch = s_0 s_1 \dots s_n$ entre s et t, avec $s_0 = s$ et $s_n = t$, est défini comme $\sum_{i=0}^{n-1} w(s_i s_{i+1})$. Le chemin ch est un L-chemin si $\sum_{i=0}^{n-1} l(s_i s_{i+1}) \leq L$. Le problème du plus court chemin restreint consiste à rechercher le L-chemin de poids minimum entre s et t dans G.*

Dans notre cas, nous recherchons une variante plus contrainte d'un L-chemin de poids minimum. Nous cherchons une variante d'un L-chemin que nous nommons un L'-chemin.

Définition 25 (L'-chemin)**.** *Un L'-chemin est un chemin $ch = s_0 s_1 \dots s_n$ ayant la propriété suivante : $\sum_{i=0}^{n-1} l(s_i s_{i+1}) = L$.*

Si pour tous les entiers U tels que $0 \leq U \leq L$ nous répondons au problème de la recherche plus court U'-chemin, nous répondons au problème du plus court chemin restreint. Ainsi, si nous répondons au problème du plus court L'-chemin en un temps polynomial, nous répondons aussi au problème du plus court chemin restreint en un temps polynomial. Cependant, le problème du plus court chemin restreint est NP-Complet[47, 81]. Il y a donc peu de chance que la recherche d'un L'-chemin soit polynomiale. Il existe des algorithmes FPTAS pour résoudre le problème du plus court chemin restreint [47, 81], mais nous ne les utiliserons pas car pour un L fixé, un L-chemin n'est pas forcément un L'-chemin.

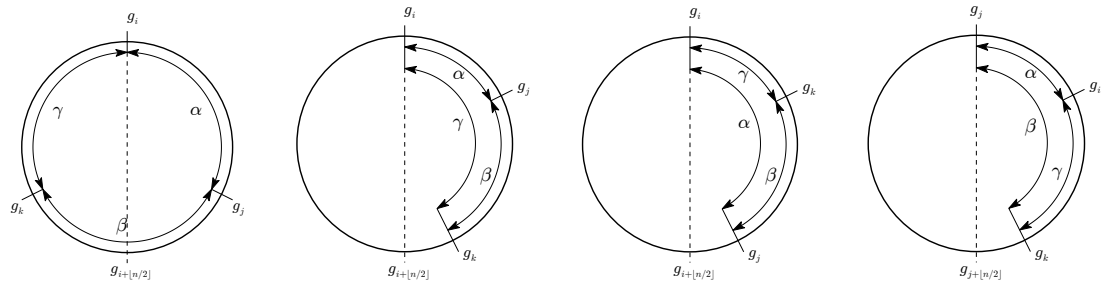
Bien que ces différents problèmes semblent proches, ils ne sont pas le nôtre. Nous pourrions obtenir une solution approchée d'un $srdr$-chemin entre deux réactions qui minimise $\overline{w}_d$ en les utilisant, mais nous avons seulement traité le cas le plus simple $\mathcal{H}$, où une réaction n'est associée qu'à un unique sommet, transformant la recherche de $srdr$-chemin en recherche de chemin sans circuit. Dans le cas général, c'est-à-dire dans $\mathcal{G}_{int}$, un $srdr$-chemin est un chemin sans circuit ni jumeaux, ce qui rend le problème de la recherche d'un $srdr$-chemin minimisant $\overline{w}_d$ plus difficile. Nous utiliserons donc nos travaux effectués sur les k-SIPs afin d'obtenir une solution approchée à nôtre problème.

Heuristique minimisant $\overline{w}_d$

Nos travaux précédents nous ont permis de rechercher, pour un entier k et deux ensembles de réactions R_1 et R_2 donnés, un k-SIP entre R_1 et R_2 dans $\mathcal{G}_{int}$. Un k-SIP sip étant un ensemble de k plus courts $srdr$-chemins, nous décidons d'approcher le calcul d'un $srdr$-chemin minimisant $\overline{w}_d$ en sélectionnant parmi les $srdr$-chemins de sip celui qui minimise $\overline{w}_d$. Nous pouvons également généraliser à la recherche de p $srdr$-chemins minimisant $\overline{w}_d$, avec $p \leq k$. Il suffit alors de sélectionner dans sip le sous-ensemble de p $srdr$-chemins minimisant $\overline{w}_d$.

3.3.5 Distances envisagées

Lors de l'élaboration de SIPPER, nous avons envisagé plus particulièrement les distances d suivantes pour pondérer $\mathcal{G}_{int}$. La première à laquelle nous nous sommes intéressés est la distance de colocalisation :



(a) $\gamma \leq \lfloor n/2 \rfloor < \alpha + \beta$ (b) $\gamma = \alpha + \beta \leq \lfloor n/2 \rfloor$ (c) $\gamma = \alpha - \beta \leq \lfloor n/2 \rfloor$ (d) $\gamma = \beta - \alpha \leq \lfloor n/2 \rfloor$

Figure 3.7 – Vérification de l'inégalité triangulaire de la distance colocalisation dans un chromosome circulaire. Trois gènes g_i, g_j et g_k sont positionnés sur le chromosome avec $\alpha = d(g_i, g_j)$, $\beta = d(g_j, g_k)$ et $\gamma = d(g_k, g_i)$. Lorsque (a) $\alpha + \beta > \lfloor n/2 \rfloor$, alors nous sommes dans un cas trivial, puisque par définition, $\gamma \leq \lfloor n/2 \rfloor$. Lorsque $\alpha + \beta \leq \lfloor n/2 \rfloor$, trois cas sont possibles : (b), (c) ou (d). Quelque soit le cas, l'inégalité triangulaire $\gamma \leq \alpha + \beta$ est vérifiée.

La distance de colocalisation entre deux gènes consiste en la différence de rang des deux gènes sur le chromosome [1]. Soit chr un chromosome contenant n gènes et soient g_i et g_j deux gènes aux positions i et j (i.e leur rang) dans chr, avec $1 \leq i, j \leq n$. Lorsque chr est linéaire, la distance de colocalisation est définie comme $d(g_i, g_j) = |i-j|$. Lorsque chr est circulaire, $d(g_i, g_j) = \min(|i-j|, n-|i-j|) \leq \lfloor n/2 \rfloor$, avec $\lfloor x \rfloor$ la partie entière inférieure de x. Quand g_i et g_j ne sont pas sur le même chromosome chr, $d(g_i, g_j) = +\infty$. Pour trois gènes g_i, g_j et g_k aux positions i, j et k sur un même chromosome, cette distance respecte l'inégalité triangulaire, même dans le cas d'un chromosome circulaire. La Figure 3.7 énumère les différents cas de figure de positionnement de gènes possibles.

Dans la continuité de la distance de colocalisation, nous avons aussi envisagé la distance intergénique :

Distance intergénique La distance intergénique [22, 27] entre deux gènes est le nombre de nucléotides sur le génome séparant deux gènes. Dans le cadre d'un chromosome circulaire, la plus petite distance possible est celle prise en compte. Dans le cadre de chromosomes multiples, la distance entre les deux gènes de chromosomes distincts sera de $+\infty$. Cette distance ne respecte pas l'inégalité triangulaire comme le montre l'exemple de la Figure 3.8. Cependant, si nous ignorons la taille des gènes, et que seule la position du premier nucléotide de chaque gène est prise en compte, il est possible de considérer la distance intergénique ainsi modifiée comme une métrique.

Distance de coexpression Dans la section 2.3.2, nous avons abordé l'étude de la similarité d'expression des gènes. Stuart *et al.* [100] utilisent comme mesure de similarité s le coefficient de corrélation de Pearson entre les niveaux d'expression de chaque couple de gènes (g_x, g_y) :

$$s = |cor(g_x, g_y)| \text{ avec } cor(g_x, g_y) \text{ la corrélation d'expression de } g_x \text{ et } g_y \text{ et } -1 \leq cor(g_x, g_y) \leq 1$$

1. Dans la littérature, il existe plusieurs définitions de colocalisation : dans certaines études il s'agit de la distance géographique entre deux gènes, qui tient compte du repliement du chromosome. Dans notre cas, nous ne tiendrons pas compte de ce repliement.

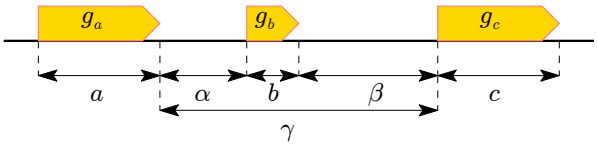


Figure 3.8 – Exemple de distance intergénique sur un chromosome. Les gènes g_a, g_b et g_c sont composés respectivement de a, b et c nucléotides. La distance intergénique entre g_a et g_b est de α nucléotides, celle entre et g_b et g_c est de β nucléotides, et celle entre g_a et g_c est de $\gamma = \alpha + b + \beta$ nucléotide. Cet exemple montre que la distance intergénique ne respecte pas l'inégalité triangulaire, car $\alpha + \beta < \gamma$.

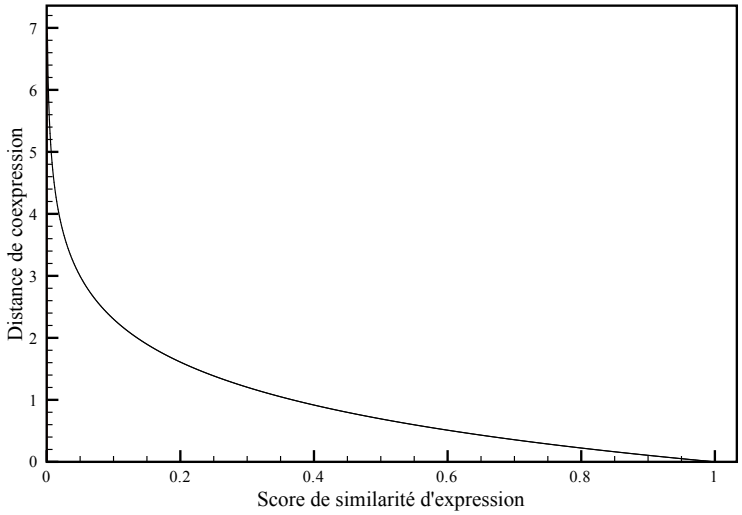


Figure 3.9 – Évolution de la distance de coexpression en fonction de la similarité d'expression.

Nous modulons cette mesure de façon similaire à celle utilisée par Zhang et Horvath [109] afin d'une part d'amplifier les fortes valeurs de corrélation d'expression et d'autre part, d'obtenir une métrique additive dans la recherche des plus courts chemins. Nous définissons ainsi la distance de coexpression d_c à partir du niveau de similarité d'expression s entre deux gènes g_x et g_y par la formule suivante :

$$d_c(g_x, g_y) = -\ln |cor(g_x, g_y)|$$

La mesure d_c code alors pour une dissimilarité. Quand g_x et g_y ont leurs niveaux d'expression fortement corrélés ou anti-corrélés, la valeur de $d_c(g_x, g_y)$ tend vers 0, tandis que quand leurs niveaux d'expression sont faiblement corrélés, la valeur de $d_c(g_x, g_y)$ tend vers $+\infty$. De plus, l'utilisation du logarithme amplifie les fortes valeurs de s qui sont celles qui nous intéressent le plus, et réduit les faibles valeurs de s comme présenté dans [100]. La Figure 3.9 illustre cette transformation. Cette distance ne respecte pas l'inégalité triangulaire, c'est-à-dire qu'il est possible d'avoir $d_c(g_x, g_y) + d_c(g_y, g_z) < d_c(g_x, g_z)$. La Table 3.1 est un exemple de cette affirmation.

Table 3.1 – Exemple de niveau d'expression de gènes montrant que la distance de coexpression n'est pas une métrique. (a) Trois gènes g_x, g_y et g_z ont leur niveaux d'expressions relevés à trois instants t_1, t_2 et t_3, formant ainsi pour chaque gène une série de relevé. (b) Il en découle la mesure de corrélation entre les niveaux d'expression de chaque couple de gènes et la distance d_c correspondante. Nous remarquons que d_c ne vérifie pas l'inégalité triangulaire car $d_c(g_x, g_y) + d_c(g_y, g_z) < d_c(g_x, g_z)$.

(a) Séries de relevés

	t_1	t_2	t_3
g_x	1	8	2
g_y	1	3	8
g_z	2	5	10

(b) Scores associés

u	v	$cor(u, v)$	$d_c(u, v)$
g_x	g_y	$-0,110$	$2,208$
g_y	g_z	$0,995$	$0,005$
g_x	g_z	$-0,011$	$4,520$

3.3.6 Forme et paramétrage des k-SIPs

Le calcul d'un k-SIP entre deux ensembles de réactions R_1 et R_2 s'effectue en calculant tout d'abord, pour un entier k donné, les plus courts $srdr$-chemins entre R_1 et R_2 comme le décrit l'algorithme `ComputeSIPs`($\mathcal{G}_{int}$,R_1, R_2, k). Quand il est projeté sur le réseau métabolique, le k-SIP(R_1, R_2) devient une collection de chaînes de réactions, chaque chaîne de réactions joignant une réaction de R_1 à une réaction de R_2. Chaque k-SIP induit un sous graphe H de $\mathcal{G}_{int}$ qui possède un coefficient de voisinage $\overline{w}_d(H)$.

Les paramètres R_1, R_2 et k sont les paramètres sur lesquels repose la flexibilité de `SIPPER`. Ils nous permettent de définir la forme des k-SIPs. Il est ainsi possible de *choisir le type de sous-réseau*. Nous pouvons en effet nous intéresser à une seule chaîne de réactions quand $k = 1$ (nous choisissons un seul 1-SIP quand plusieurs ont le même score), ou bien à des chaînes de réactions alternatives quand $k > 1$, tout en jouant sur la cardinalité de R_1 et R_2. Il est ainsi possible, lorsque R_1 est un singleton et R_2 un ensemble de plusieurs réactions, d'obtenir des chaînes arborescentes de réactions, c'est-à-dire qu'à partir d'une réaction, il est possible d'atteindre plusieurs réactions distinctes à l'aide de l'ensemble des k chaînes de réactions calculées. Il est aussi possible de *choisir la taille du sous-réseau*. La variation de k nous permet de réaliser une étude progressive des chemins joignant R_1 et R_2. Nous pouvons savoir s'ils varient très peu (i.e. ils ont beaucoup de parties communes) ou beaucoup (i.e. ils ont peu de parties communes).

Il est tout à fait possible de *filtrer les k-SIPs par rapport à la distance entre unités catalytiques*. Un k-SIP(R_1, R_2) est une collection de plus courts chemins allant des réactions de R_1 à celles de R_2 dans $\mathcal{G}_{int}$ qui dispose d'une mesure $\overline{w}_d$. Le coefficient de voisinage $\overline{w}_d$, et la densité génomique d_G quand nous travaillons sur la version "gène" de `SIPPER`, permettent de filtrer les k-SIPs obtenus en fixant un seuil de pertinence. Nous pouvons ainsi chercher, de façon inexacte, des k-SIPs qui minimisent $\overline{w}_d$. Quand nous les projetons sur le réseau métabolique, nous obtenons un sous-réseau qui joint les réactions de R_1 à celles de R_2 avec une plus petite distance d moyenne entre les unités catalytiques impliquées dans la catalyse des réactions. Ainsi en choisissant les paramètres énumérés précédemment de `SIPPER`, il est possible d'identifier des sous-réseaux de types, tailles et coefficients de voisinage désirés. Cette analyse peut être effectuée à petite ou grande échelle, en s'intéressant aux réactions, ou aux chemins, ou aux sous-graphes.

3.4 Comparaison de SIPPER avec d'autres approches

Dans l'article [78], Ogata *et al.* ont proposé une méthode qui extrait des ensembles d'enzymes, appelés FRECs, qui catalysent des réactions successives et qui sont encodées par des gènes proches sur le génome. Dans cette approche, les auteurs ont représenté les informations métaboliques et génomiques sous la forme suivante : le réseau métabolique et le génome sont vus comme deux graphes non-orientés distincts G_1 et G_2. Dans G_1, les sommets représentent les enzymes qui catalysent les réactions ; deux enzymes sont alors connectées lorsque les réactions qu'elles catalysent chacune partagent des métabolites communes. Dans G_2, les sommets sont les gènes, et une arrête lie deux gènes adjacents dans le génome. Une correspondance entre un sommet de G_1 et un sommet G_2 est établie si l'enzyme associée au sommet de G_1 a le même E.C. number [2] (i.e. un identifiant) que le gène associé au sommet de G_2. Quand un sommet de G_1 (respectivement G_2) correspond à plusieurs sommets dans G_2 (respectivement G_1), ce sommet est copié plusieurs fois afin d'avoir une bijection entre les sommets de G_1 et ceux de G_2. Cela correspond aux couplages (gène, réaction) dans $\mathcal{G}_{int}$. Les topologies de G_1 et G_2 sont ensuite comparées à l'aide d'un clustering progressif. Au départ, chaque sommet forme un cluster. Ensuite, étant donné deux entiers Gap_1 et Gap_2, deux clusters C_1 et C_2 sont fusionnés si, dans G_1, il existe un chemin entre un des sommets de C_1 et un sommet de C_2 dont la longueur est inférieure au seuil $Gap_1 + 1$ et, dans G_2 il existe un chemin entre un des sommets de C_1 et un sommet de C_2 dont la longueur est inférieure au seuil $Gap_2 + 1$. Cette opération est répétée jusqu'à ce qu'il ne soit plus possible de fusionner de clusters. Par rapport à SIPPER, cette méthode considère que les réactions métaboliques sont toutes réversibles. L'utilisation des enzymes à la place des réactions peut créer des raccourcis qui font que des enzymes s'enchainent sans qu'il existe de lien entre les réactions qu'elles catalysent. En utilisant la version "enzyme" de SIPPER, il est possible d'envisager une approche similaire à la méthode d'Ogata *et al.* Il suffit de pondérer le graphe intégré $\mathcal{G}_{int}$ par la distance de colocalisation et de rechercher ensuite tous les 1-SIPs dans $\mathcal{G}_{int}$. Chaque 1-SIP sip étant un ensemble singleton, nous appelons ch_{sip} le chemin de sip. Nous sélectionnons donc ensuite chaque 1-SIP sip qui respecte les conditions suivantes : $w_d(ch_{sip}) < Gap_1$ et $||ch_{sip}|| < Gap_2$. En regroupant ensuite ensemble les 1-SIPs ayant au moins un sommet commun, il est possible d'obtenir des regroupement de sommets de $\mathcal{G}_{int}$ qui ressemblent à des FRECs. Il est aussi possible de mettre en place une méthode similaire à la généralisation de la méthode d'Ogata *et al.* par Nakaya *et al.* [76]. Dans cette généralisation, les auteurs utilisent comme exemple la recherche des clusters d'enzymes qui catalysent des réactions successives et qui sont encodées par des gènes à la fois proches sur le génome et dont l'expression est similaire. Il suffit d'utiliser la version "gène" de SIPPER avec une distance regroupant à la fois la similarité d'expression des gènes et la distance de colocalisation, et d'appliquer le même procédé que pour approcher la méthodes d'Ogata *et al.*

De leur côté, Boyer *et al.* [20] ont utilisé une modélisation qui a servi de base à la nôtre. En effet, bien que les auteurs utilisent un multi-graphe plutôt qu'un graphe, l'information génomique et métabolique est prise en compte de manière similaire : chaque sommet du multi-graphe est un couple (gène, réaction) obtenu de la même façon que ceux de $\mathcal{G}_{int}$. La différence des modèles est que Boyer *et al.* utilisent un multi-graphe non orienté et non pondéré, tandis que nous utilisons un graphe simple orienté et pondéré. Par ces différences, Boyer *et al.* peuvent cumuler plus facilement d'autres informations, telles que des informations trancriptomiques ou d'interaction de protéines, en les 'superposant' au détriment d'une structure de données plus lourde à stocker et analyser qui est liée à la multiplication des arrêtes du multi-graphe. De notre côté, nous devons les coder sous la forme de poids sur les arcs. Ensuite, vient la recherche d'entités biologiques. Boyer *et al.* recherchent des composantes connexes communes (CCC), qui

2. Une définition de l 'EC number est disponible à la section 4.2.1

sont basées sur la connectivité des entités biologiques entre-elles. Elles expriment des groupes d'entités biologiques qui sont toujours connexes dans toutes les vues du système. De notre côté, nous recherchons des k-SIPs, chemins orientés qui minimisent une distance codant pour une hypothèse ou une propriété biologique donnée. L'intérêt des CCCs par rapport aux k-SIPs est qu'il n'y a pas besoin de définir de sommets de départ et d'arrivé pour un chemin, qui constituent des entrées et sorties du système biologique. Cependant les CCCs ne se basent que sur la notion de connectivité aux sein du multi-graphe étudié et ne tiennent pas compte de la qualité des liens (i.e. le poids des arcs/arêtes) entre réactions, ni de la notion de précédence lors d'un enchaînement de réactions. De ce point de vue, l'approche de Boyer *et al.* semble moins flexible que SIPPER, même si elle peut s'appliquer à plus de données biologiques.

Simeonidis *et al.* [98] ont mis en place une approche de programmation linéaire pour calculer des plus courts chemins non pondérés entre des paires d'enzymes sur le réseau métabolique. Ils ont ensuite comparé (a) la distance obtenue sur le réseau métabolique entre deux enzymes, avec (b) la distance, en nombre de paires de bases, entre leur gènes correspondants sur le génome, et aussi avec (c) une distance fonctionnelle entre enzymes, déduite de leur classification fonctionnelle (basée sur l'EC number). Les auteurs en ont conclu que les distances en (a) et (b) sont corrélées, mais pas les distances (a) et (c). De telles comparaisons peuvent-être reproduites avec SIPPER. En utilisant la version "gène" avec la distance (b) pour la première comparaison, et la version "enzyme" et la distance (c) pour la seconde comparaison, l'étude de la distribution des valeurs de $\overline{w}_d$ des 1-SIPs entre les réactions nous permettrait de parvenir aux mêmes conclusions. Il serait même possible de faire une analyse simultanée des trois distances en combinant les distances (b) et (c) en une seule distance.

En choisissant de paramètres proches des méthodes originales, SIPPER est capable de reproduire ces méthodes, ce qui illustre sa flexibilité.

3.5 Conclusion

Nous avons mis en place SIPPER, une méthode s'appliquant sur les systèmes bactériens qui :

1. utilise conjointement un génome ou un ensemble d'enzymes et un réseau métabolique sous la forme d'un graphe intégré $\mathcal{G}_{int}$ et le pondère à l'aide d'une distance qui dépend d'une propriété biologique (quelques unes possibles sont présentées),
2. puis pour un entier k donné, recherche des k-SIPs, un ensemble de k plus courts $srdr$-chemins (i.e. chemins sans répétition de réactions) dans $\mathcal{G}_{int}$ allant d'un ensemble de réactions donné à un autre,
3. et définit leur intérêt en utilisant principalement la mesure $\overline{w}_d$ des k-SIPs et de façon plus ponctuelle, la mesure de densité génomique d_G.

Nous avons également montré que la recherche d'un $srdr$-chemin minimisant $\overline{w}_d$ semble difficile, en étudiant un cas simple de $\mathcal{G}_{int}$ où chaque réaction n'est associée qu'à un sommet. Nous avons proposé d'approcher ces $srdr$-chemins minimisant $\overline{w}_d$ en sélectionnant un sous-ensemble de $srdr$-chemins d'un k-SIP. Dans le chapitre suivant, nous allons appliquer SIPPER sur *E. coli*, l'organisme bactérien de référence pour les tests biologiques, afin de vérifier si notre méthode est cohérente vis à vis des méthodes décrites précédemment.

CHAPITRE 4

Applications et résultats biologiques

4.1 Introduction

Ce chapitre reprend les travaux effectués dans [17] et les enrichit. Dans cet article, nous avons appliqué la méthode décrite dans le chapitre précédent sur les données de la bactérie unichromosomale *Escherichia coli* afin d'étudier l'implication du voisinage de gènes dans le réseau métabolique bactérien avec `SIPPER`. Nous avons choisi *E. coli* en particulier, car c'est l'organisme procaryote le plus étudié, et c'est donc le candidat idéal pour évaluer notre méthode. Nous allons donc d'abord présenter les données prises en compte lors de cette étude, puis les méthodes d'évaluation des résultats qui ont été mises en place. Enfin nous allons présenter et analyser les résultats biologiques obtenus lors de l'étude de l'implication du voisinage de gènes dans le réseau métabolique bactérien et, pour allez plus loin, également ceux de l'étude de l'implication de la similarité d'expression des gènes dans le réseau métabolique bactérien.

4.2 Jeux de données

Nous nous sommes intéressés à la souche *K12 MG 1655* d'*E. coli*. Cette souche de référence d'*E. coli* est la plus étudiée et constitue ainsi l'organisme de référence des procaryotes. Elle est donc très utilisée à des fins d'évaluation ou d'apprentissage. Dans nos travaux, nous allons utiliser les données génomiques, métaboliques et transcriptomiques d'*E. coli*.

4.2.1 Les données génomiques

Le génome d'*E. coli* utilisé au cours de cette étude provient de la base NCBI/GenBank [16]. Cette base est intéressante car elle contient tous les génomes publics publiés et annotés des organismes séquencés. Ainsi, pour chacun des gènes d'un génome, la position de celui-ci est connue et les protéines qu'il génère sont annotées et classifiées fonctionnellement par un code nommé *EC number*. La version du génome que nous avons utilisée est celle du 31 mars 2008. Ce génome consiste en un seul chromosome circulaire de 4242 gènes. Cette forme de génome unichromosomal se retrouve chez la majorité des bactéries.

4.2.2 Les données métaboliques

Le réseau métabolique d'*E. coli* utilisé provient de la base de données de voies métaboliques KEGG PATHWAYS [57]. Les données concernant *E. coli* sont reconnues par l'identifiant *eco* au sein de la base.

Nous avons utilisé la version du 21 octobre 2008. À l'époque, cette base fournissait le réseau métabolique comme un ensemble de cartes de voies métaboliques, aussi appelées cartes métaboliques. Le réseau métabolique global a été reconstruit en prenant, dans chaque carte, les réactions et les métabolites présents. Une réaction et un métabolite du réseau métabolique sont alors connectés dans le réseau métabolique global s'ils sont connectés dans une des cartes métaboliques de KEGG [1]. Afin de limiter le problème des métabolites annexes qui court-circuitent les enchaînements de réactions (voir section 2.2.1, page 10), les métabolites participant à plus de 25 réactions différentes ont été retirés du réseau métabolique. Chaque réaction contenue dans les cartes métaboliques de KEGG est annotée. Les enzymes qui la catalysent sont identifiables grâce aux EC numbers qui lui sont associés. Nous avons donc obtenu de cette façon un réseau métabolique composé de 2971 métabolites impliqués dans 1131 réactions catalysées par 647 enzymes différentes.

Une base de données alternative pour se procurer des voies, cartes et réseaux métaboliques est la base Metacyc [24]. Cette base fournit, pour un organisme donné, une multitude d'informations biologiques, en particulier les informations génomiques et métaboliques, connues à son propos.

4.2.3 Les données de transcriptomique/d'expression

Il existe plusieurs bases de données pour se procurer des données d'expression de gènes. La difficulté est que ces données dépendent de conditions expérimentales particulières. Les données d'expression des gènes d'*E. coli* proviennent de la base Gene Expression Omnibus (GEO) [36, 11]. Nous avons utilisé les données issues des expériences réalisées dans [92], plus particulièrement celles de la croissance anaérobie d'une population de bactéries *E. coli* (puce à ADN de 4290 ORFs [2]) sur un média de croissance M9, une solution saline qui est proche de l'eau de mer, avec du glucose comme nutriment (référence GDS2588 dans la base GEO).

4.3 Recherche de k-SIPs

Dans ce chapitre, nous allons étudier deux hypothèses biologiques à partir desquelles nous allons appliquer `SIPPER` en tant que méthode exploratoire, afin 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. Nous allons générer deux instances de $\mathcal{G}_{int}$, nommées $\mathcal{G}_{col}$ et $\mathcal{G}_{coexp}$, puis calculer, pour chacune de ces instances et pour k de 1 à 10, l'ensemble des k-SIPs entre singletons de réactions. Nous les analyserons ensuite à la fois d'un point de vue génomique et métabolique pour vérifier si chaque hypothèse formulée nous permet d'obtenir des résultats cohérents par rapport à la littérature, et ainsi valider `SIPPER`.

1. Au moment de la rédaction de ce manuscrit, la reconstruction n'est plus nécessaire car une carte globale est directement disponible sur le site de KEGG.
2. Un ORF est une séquence d'ADN codant pour un gène.

4.4 Évaluation des résultats

4.4.1 Données d'évaluation

Les k-SIPs mis en valeur par notre approche SIPPER seront confrontés à diverses connaissances de la littérature pour évaluer l'apport biologique de SIPPER.

Génomes aléatoires

Afin de tester l'impact de l'ordre des gènes sur le génome dans la découverte de k-SIPs dans $\mathcal{G}_{col}$, nous avons généré des génomes aléatoires en mélangeant le génome original d'*E. coli*. Le mélange consiste à considérer le génome comme une liste de gènes dans laquelle les gènes sont permutés de façon uniforme en échangeant, pour chaque élément de la liste du dernier jusqu'au second, l'élément actuel avec un autre élément pris aléatoirement dans la liste selon un tirage équiprobable. Lors de nos tests, nous avons généré dix génomes aléatoires afin de tester l'influence de l'ordre des gènes d'un génome sur les k-SIPs.

Réseaux métaboliques aléatoires

Afin de tester l'impact de l'organisation du réseau métabolique dans la découverte de k-SIPs dans $\mathcal{G}_{int}$, nous avons généré des réseaux métaboliques aléatoires en nous inspirant des travaux présentés dans [72]. Le réseau métabolique d'*E. coli* est représenté par un graphe de réactions orienté $\mathcal{M}$. Nous avons constitué la liste O des sommets d'origine de chaque arc de $\mathcal{M}$ et la liste D des sommets de destination de chaque arc telles que le rang i dans O et D soit le rang du i-ième arc de $\mathcal{M}$ selon l'ordre lexicographique. Pour obtenir un réseau métabolique aléatoire $\mathcal{M}'$, les listes O et D sont mélangées de façon uniforme de la même façon que dans le cas des génomes aléatoires. Deux nouvelles listes O' et D' sont ainsi obtenues avec la propriété suivante : le sommet o'_j au rang j dans O' et le sommet d'_j au rang j dans D' forment ainsi l'arc $o'_j d'_j$ du réseau métabolique aléatoire $\mathcal{M}'$. S'il existe au moins une boucle (i.e. un arc xx) dans $\mathcal{M}'$, un nouveau mélange de O et D est effectué jusqu'à ce que $\mathcal{M}'$ soit sans boucle. Lors de nos tests, dix réseaux métaboliques aléatoires ont été générés afin de tester l'influence de la topologie du métabolisme sur les k-SIPs.

Gènes essentiels

Les gènes essentiels [10] sont des points importants du génome. L'absence de l'un d'entre eux dans le génome d'un organisme rend cet organisme non viable dans un environnement favorable à sa croissance. La base de données PEC [58] répertorie 303 gènes essentiels pour *E. coli*. Parmis ceux-ci, 133 gènes catalysent des réactions métaboliques.

Notre objectif est de vérifier quelle proportion de ces 133 gènes intervient dans les k-SIPs et si leur fréquence d'apparition dans les k-SIPs est très importante ou non.

Opérons

Un opéron est une unité de transcription, un groupe de gènes consécutifs dont l'expression est régulée par un ou plusieurs facteurs de transcription [52]. Il est constitué d'au moins deux gènes sur le génome. Nous considérons ici les opérons dits métaboliques qui sont des unités de transcription constituées d'au moins deux gènes qui catalysent des réactions métaboliques. La base RegulonDB [90] dans

sa version 6.3, datée du 30 janvier 2009, référence 833 opérons connus chez *E. coli*. Parmi ceux-ci, seuls 135 (16,2%) correspondent à des opérons métaboliques. Nous noterons cet ensemble d'opérons métaboliques comme l'ensemble $Operons$. Lorsque nous comparons un k-SIP avec un opéron, nous comparons l'ensemble des gènes du k-SIP avec l'ensemble des gènes de l'opéron.

De manière générale, nous noterons qu'il existe aussi des groupes d'opérons qui sont conservés de façon contigüe dans une majorité de génomes (plus de 50% des organismes testés). Ce sont les über-opérons [26] qui possèdent un intérêt évolutif et fonctionnel.

Modules métaboliques de KEGG

Les modules de KEGG sont de petites parties de voies métaboliques. Ils sont définis manuellement comme la combinaison de complexes moléculaires, de réactions consécutives, d'unités fonctionnelles, régulatrices ou phylogénétiques [55]. Ces modules sont définis de façon globale pour l'ensemble des voies métaboliques connues tous organismes confondus. Nous avons donc sélectionné ceux qui existent dans *E. coli* et qui sont composés d'au moins deux réactions enzymatiques. Nous obtenons ainsi un ensemble de 99 modules métaboliques de KEGG sélectionnés parmi 689 (14,39%). Nous noterons cet ensemble de modules métaboliques comme l'ensemble *Modules*.

Ces modules sont l'équivalent des Pathways disponibles dans la base Metacyc [24].

L'application de `SIPPER` nous permet d'obtenir des k-SIPs dont nous allons étudier la signification. Nous avons donc besoin de mesures pour évaluer sur les données proposées précédemment.

4.4.2 Mesures de qualité

Afin d'évaluer chacun de ces k-SIPs obtenus lors de nos applications, nous allons les comparer avec des ensembles de gènes ou de réactions. Un k-SIP n'est pas un ensemble de gènes ou de réactions, mais il en contient. Ainsi, lorsque nous le comparerons avec un ensemble de gènes F, nous comparerons l'ensemble des gènes qui interviennent dans le k-SIP à F. De même, lorsque nous comparerons un k-SIP avec un ensemble de réactions R, nous comparerons l'ensemble des réactions qui interviennent dans le k-SIP à R. Pour comparer de tels ensembles, nous utiliserons les mesures de similarité suivantes : la *couverture* et la *mesure de Jaccard*.

La couverture

Étant donnés deux ensembles non vides A et B de même type, *la couverture* de A par B est définie comme :

$$\textit{Couverture}(A, B) = \frac{|A \cap B|}{|A|}$$

Cette mesure permet d'évaluer la proportion de A qui est aussi dans B. Elle prend des valeurs comprises entre 0 et 1. Lorsque $\textit{Couverture}(A, B) = 1$, tous les éléments de A sont dans B et nous dirons alors que *A est couvert totalement par B*. Lorsque $\textit{Couverture}(A, B) = 0$, aucun élément de A est dans B. Nous dirons que *A n'est pas couvert par B*. Par abus de langage, lorsque nous mesurons la couverture d'un opéron op (respectivement d'un module métabolique m) par un k-SIP sip, nous mesurons la couverture de l'ensemble des gènes de op (respectivement l'ensemble des réactions de m) par l'ensemble des gènes (respectivement l'ensemble des réactions) intervenant dans sip. Nous utilisons alors la notation suivante : $\textit{Couverture}(op, sip)$ (respectivement $\textit{Couverture}(m, sip)$).

La mesure de Jaccard

Étant donnés deux ensembles non vides A et B de même type, *la mesure de Jaccard* entre A et B est définie comme :

$$\textit{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Cette mesure, aussi appelée la couverture mutuelle entre A et B, permet de mesurer à la fois la proportion de A qui est dans B et la proportion de B qui est dans A. Elle prend des valeurs comprises entre 0 et 1. Lorsque $\textit{Jaccard}(A, B) = 1$, A et B sont identiques et nous dirons que A *correspond exactement à* B, et lorsque $\textit{Jaccard}(A, B) = 0$, A *et* B *sont distincts*.

Par abus de langage, lorsque nous calculons la mesure de Jaccard entre un opéron op et un k-SIP sip, nous calculons la mesure de Jaccard entre l'ensemble des gènes de op et l'ensemble des gènes intervenant dans sip. Nous utilisons alors la notation suivante : $\textit{Jaccard}(op, sip)$. De plus, nous désignons par *pertinence opéronique* P_o de sip le résultat de la formule suivante :

$$P_o = \max_{op \in \textit{Operons}} \{\textit{Jaccard}(op, sip)\}$$

De même lorsque nous calculons la mesure de Jaccard entre un module métabolique de KEGG m et un k-SIP sip, nous calculons la mesure de Jaccard entre l'ensemble des réactions de m et l'ensemble des réactions intervenant dans sip. Nous utilisons alors la notation suivante : $\textit{Jaccard}(m, sip)$. Nous désignons également la *pertinence modulaire* P_m de sip comme étant :

$$P_m = \max_{m \in \textit{Modules}} \{\textit{Jaccard}(m, sip)\}$$

La p-valeur

La *p-valeur* est une valeur statistique qui quantifie la qualité d'une mesure ou d'un échantillon. Dans le cas d'un échantillon E, cette valeur décrit de façon équivalente la probabilité de E à appartenir une catégorie d'intérêt, ou la probabilité un d'observer un résultat au moins aussi extrême que celui de l'échantillon [86].

La p-value de E est calculée en supposant l'hypothèse nulle H_0 : les données observées sont dues au hasard. Si la valeur de cette probabilité est inférieure à un seuil, par exemple de 0,05, qui est une valeur couramment utilisée, alors l'hypothèse H_0 est rejetée (avec un taux d'erreur de 5%). La mesure observée sur l'échantillon est ainsi significative d'un point de vue statistique, puisqu'elle est difficile à obtenir par hasard. Dans notre cas, la p-valeur sert à évaluer si un ensemble de gènes ou de réactions est obtenu de façon plus significative que par hasard.

Afin de calculer la p-valeur de F, un ensemble de gènes donné, nous avons utilisé l'outil `GO::TermFinder` [21] qui se base sur les connaissances contenues dans les bases de connaissances au format Gene Ontology [19]. Cet outil calcule la p-valeur de F en déterminant la probabilité d'observer F par un tirage aléatoire sans remise des gènes contenus dans un base de connaissance au format Gene Onthology (i.e. tirage suivant une distribution hypergéométrique). Dans le cadre de nos applications sur *E. coli*, la base de connaissances Gene Ontology utilisée est issue d'Ecocyc [59] à la date du 02 septembre 2009.

Corrélation linéaire

La *corrélation linéaire* de plusieurs éléments d'un échantillon entre deux ou plusieurs critères consiste à étudier l'intensité de la liaison qui peut exister entre les critères dans l'échantillon donné. Cette liaison

est une relation affine. Dans le cas de deux critères, il s'agit de la *régression linéaire* qui est obtenue à partir du *coefficient de corrélation linéaire*.

Le *coefficient de corrélation linéaire* est compris entre -1 et 1 et est égal au rapport de la covariance des critères et du produit non nul des écarts-types de chacun des critères. Dans le cas de deux critères x et y dans l'échantillon F, nous avons $X(x_1, \ldots, x_n)$ et $Y(y_1, \ldots, y_n)$ les séries de valeurs associées aux critères x et y et nous calculons r_p, le coefficient de corrélation linéaire, de la façon suivante :

$$r_p = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} \text{ avec } \sigma_{XY} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x}) \times (y_i - \overline{y}), \text{ la covariance entre } X \text{ et } Y,$$

$$\sigma_X = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}, \text{ l'écart type de } X,$$

$$\sigma_Y = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2}, \text{ l'écart type de } Y,$$

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \text{ la moyenne de } X,$$

$$\text{et } \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i, \text{ la moyenne de } Y.$$

La *régression linéaire* est une fonction linéaire qui représente au mieux la relation qui existe entre les valeurs (x_i, y_i) que peut prendre l'échantillon F. Elle s'écrit sous la forme d'un équation affine $\widehat{y} = a\widehat{x} + b$. Les valeurs de a et b sont obtenues à l'aide de la méthode des moindres carrés, qui permet de minimiser l'écart qui existe entre chaque (x_i, y_i) de F et chaque valeur estimée $(\widehat{x}, \widehat{y})$:

$$a = \frac{\sigma_{XY}}{\sigma_X^2} \text{ et } b = \overline{y} - a\overline{x}$$

La *droite de régression linéaire* est la représentation graphique de la régression linéaire. Elle permet de visualiser sur un graphe, où chaque point est un élément de l'échantillon F selon les critères x et y, à quel point x et y sont proches de l'estimation représentée par la régression linéaire. Le *coefficient de détermination*, aussi appelé R^2, mesure la force de la liaison qui existe entre cette droite et F. Le R^2 est calculé de cette façon :

$$R^2 = 1 - \frac{SS_{err}}{SS_{tot}} \text{ avec } SS_{err} = \sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$$

$$SS_{tot} = \sum_{i=1}^{n}(y_i - \overline{y})^2$$

$$\widehat{y}_i = ax_i + b$$

La valeur de R^2 exprime la part de la variance de X qui est expliquée par la liaison de X avec Y. Ainsi, plus le R^2 s'approche de 1, plus les valeurs observées de (x_i, y_i) sont proches de celles obtenues par l'équation de la droite de corrélation.

Test bilatéral de Wilcoxon

Le test bilatéral de Wilcoxon est un test statistique non paramétrique. Il vérifie si deux échantillons indépendants E_1 et E_2, avec $|E_1| = n$ et $|E_2| = m$, suivent une même distribution selon un critère c (hypothèse H_0) en comparant leur rang moyen sur le critère c lorsqu'ils sont regroupés dans un même ensemble $E = E_1 \cup E_2$. En effet, si E_1 et E_2 suivent la même distribution, alors le rang moyen des éléments de E_1 dans E doit être très proche de celui des éléments de E_2.

Dans E, soient $S_1, \ldots, S_n$ les rangs des éléments de E_1 et $T_1, \ldots, T_m$ ceux de E_2. Le test de Wilcoxon va évaluer la probabilité que la distribution des rangs S_i, et donc celle des rangs T_j, soit régulière dans E, c'est-à-dire évaluer E_1 et E_2 sous l'hypothèse H_0. Pour cela, nous calculons W_1 la somme des rangs de E_1 et nous fixons un seuil $\alpha = 0,05$ dans notre cas, puis nous cherchons la valeur W_α pour α donné et $n + m$ dans la table des valeurs critiques du test de Wilcoxon [75]. L'hypothèse H_0 est rejetée avec un taux d'erreur de α si W_1 est hors de l'intervalle $I_\alpha = [W_{\alpha/2}, n(n + m + 1) - W_{\alpha/2}]$. Lorsque E_1 ou E_2 sont grands (>10), et c'est notre cas, le test bilatéral de Wilcoxon est approché en vérifiant que Z suit une loi normale $\mathcal{N}(0, 1)$ avec :

$$Z = \frac{W_1 - n(n + m + 1) \times \frac{1}{2}}{\sqrt{nm(n + m + 1) \times 12}}$$

Nous utiliserons l'outil statistique `R` afin de réaliser ce test statistique. Afin de simplifier la lecture de la validité de cette hypothèse et ainsi accepter ou rejeter H_0, nous utiliserons la p-valeur fournie par `R`, qui est une réécriture du test d'appartenance de W_1 à I_α. Lorsque la p-valeur est inférieure à α, W_1 est dans I_α et l'hypothèse H_0 ne peut-être rejetée (avec un taux d'erreur de α).

Test du χ^2

Le test du χ^2 est un test statistique non paramétrique. Il permet de vérifier si la fréquence d'apparition d'éléments constituant un échantillon E_1 est similaire soit à une loi de probabilité, soit à celle des éléments d'un autre échantillon E_2 (test d'homogénéité). Dans notre cas, nous utiliserons le test d'homogénéité avec H_0 : "E_1 est similaire à E_2". Afin d'expliquer à quoi cela correspond, nous considérons deux échantillons E_1 et E_2. Les individus de chaque échantillon sont répartis dans des classes C_1 et C_2. Nous obtenons donc la table de contingence suivante :

	Effectif		
Classes	E_1	E_2	Total
C_1	N_{11}	N_{12}	$N_{1\bullet} = \sum_{i=1}^{2} N_{1i}$
C_2	N_{21}	N_{22}	$N_{2\bullet} = \sum_{i=1}^{2} N_{2i}$
Total	$N_{\bullet 1} = \sum_{i=1}^{2} N_{i1}$	$N_{\bullet 2} = \sum_{i=1}^{2} N_{i2}$	$N_{\bullet\bullet}$

Le χ^2 d'homogénéité correspond alors à ce qui suit :

$$\chi^2_{(n-1)\times(p-1)} = \sum_{1 \le i \le p, 1 \le j \le n} \frac{(N_{ij} - \widehat{N}_{ij})^2}{\widehat{N}_{ij}}, \text{ avec } \widehat{N}_{ij} = \frac{N_{i\bullet} \times N_{\bullet j}}{N_{\bullet\bullet}}$$

où n est le nombre d'échantillons et p le nombre de classes. Le produit $(n - 1) \times (p - 1)$ s'appelle le *degré de liberté* du χ^2.

Le test du χ^2 correspond à une différence des moindres carrés entre les valeurs des échantillons observés, représentées par leurs valeurs N_{ij}, et les valeurs de la distribution homogène de ces échantillons,

représentées par les valeurs $\widehat{N}_{ij}$, pondérée par ces valeurs "homogènes". Il faut ensuite se référer à la table des valeurs critiques du χ^2 [75] pour savoir si la valeur de χ^2 permet de rejeter ou non l'hypothèse H_0 avec un taux d'erreur que nous fixons à 5%.

Comme pour le test bilatéral de Wilcoxon, nous utiliserons les fonctions intégrées au logiciel `R` pour réaliser le test du χ^2.

4.5 Implication du voisinage de gènes dans les enchaînements de réactions

Comme mentionné dans le chapitre 2, les gènes voisins sur le génome sont fortement liés fonctionnellement [44, 87].Nous nous sommes donc focalisés sur la découverte de k-SIPs biologiquement significatifs dont les gènes sont proches sur le génome.

4.5.1 Génération de $\mathcal{G}_{col}$

Afin d'étudier le liens entre le voisinage de gènes et les chemins dans les réseaux métaboliques bactériens, nous allons appliquer la méthode décrite au chapitre précédent sur les données d'*E. coli*. Nous construisons donc le modèle utilisant les données d'*E. coli* et nous notons ce modèle $\mathcal{G}_{col}$. Nous y fixons l'ensemble des gènes du génome d'*E. coli* comme l'ensemble d'unités catalytiques. Ainsi l'ensemble des sommets de $\mathcal{G}_{col}$ est un ensemble de couples (gène, réaction), où chaque gène est associé à chaque réaction qu'il catalyse via les enzymes qu'il encode. Afin d'inclure la notion de voisinage de gènes sur le génome à $\mathcal{G}_{col}$, nous avons choisi la distance de colocalisation, qui est décrite dans la section 3.3.5, page 44. Ainsi chaque arc de $\mathcal{G}_{col}$ est pondéré par la différence, sur le génome, entre la position des gènes qui sont à ses extrémités. L'instance $\mathcal{G}_{col}$ obtenue est alors composée de 2343 sommets et 13288 arcs qui encodent pour 779 gènes dits métaboliques (18,36% du génome). Ces gènes catalysent dans le réseau métabolique 1049 réactions (92,75% des réactions d'*E. coli*) via 558 enzymes.

4.5.2 Résultats

Pour chaque valeur de k comprise entre 1 et 10, nous avons calculé l'ensemble k-$\mathcal{SIP}_{col}$ composé des k-SIPs dans $\mathcal{G}_{col}$ entre chaque couple de réactions. Nous avons utilisé pour cela l'algorithme `computeSIP`($\mathcal{G}_{col}$, R_1, R_2) décrit page 40, R_1 et R_2 étant alors des singletons de réactions et $R_1 \neq R_2$. Dans la suite de ce chapitre, lorsque nous évoquerons un k-SIP sans précision, nous évoquerons un k-SIP entre couple de réaction. La cardinalité, c'est-à-dire le nombre d'éléments, de chaque k-$\mathcal{SIP}_{col}$ est de 439382. En moyenne, un 1-SIP et un 10-SIP contiennent 11,8 ($\pm 4,5$) gènes et 13,8 ($\pm 4,7$) gènes ainsi que 13,9 ($\pm 5,6$) et 17,6 ($\pm 6,2$) réactions (voir Table 4.1 pour les k intermédiaires). Cette variation de résultats montre que l'ajout de chemins alternatifs (i.e. de 1 à 10) impacte faiblement sur le nombre de gènes et de réactions utilisés dans les k-SIPs. Cette observation s'explique par le fait que les chemins alternatifs qui composent un k-SIP diffèrent très peu les uns des autres.

Vu le nombre conséquent de k-SIPs, nous allons vérifier s'ils sont tous biologiquement cohérents. Nous confrontons chacun d'entre eux à la connaissance génomique contenue dans Ecocyc [59] grâce au logiciel `GO::TermFinder`. Nous analyserons ensuite, pour chaque k entre 1 et 10, l'ensemble k-$\mathcal{SIP}_{col}$ afin de répondre à la question : "Est-il possible de mettre en évidence les gènes essentiels à l'aide des k-SIPs ?" De même, en s'intéressant à une partie d'entre eux à l'aide du coefficient de voisinage $\overline{w}_d$ et de la densité génomique d_G (voir section 3.3.2), nous cherchons à répondre aux questions :

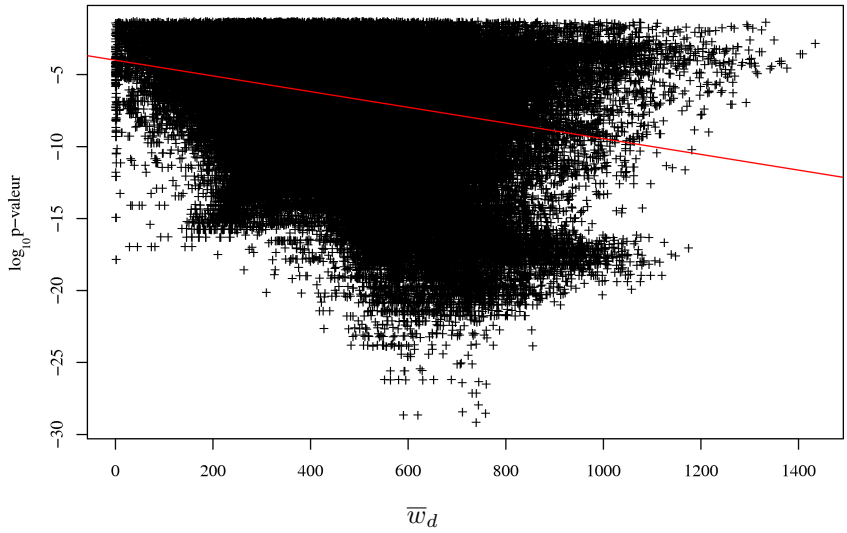


Figure 4.1 – Nuage de points présentant les 1-SIPs de 1-$\mathcal{SIP}_{col}$ sous la formes de couples ($\overline{w}_d$,p-valeur). Chaque point représente un 1-SIP. La droite rouge est la droite de régression linéaire du nuage de points. Son R^2 est de 0,074

"Pouvons nous déterminer des opérons grâce aux k-SIPs ?" puis "Pouvons nous déterminer des modules métaboliques à l'aide des k-SIPs ?"

4.5.2.1 Les k-SIPs de $\mathcal{G}_{col}$ constituent des groupes de gènes pertinents biologiquement

Nous calculons la p-valeur, dans la base de connaissance Ecocyc, de chaque k-SIP de k-$\mathcal{SIP}_{col}$ avec `GO::TermFinder` afin de déterminer si le k-SIP est biologiquement pertinent. Pour $k = 1$, 99,22% des 1-SIPs ont une p-valeur significative (i.e. $< 0,05$). Ce résultat montre que les 1-SIPs obtenus ne sont pas le fruit du hasard. Nous observons le même résultat pour les autres valeurs de k : les gènes qui participent à un enchaînement de réactions sont significatifs ensemble, ce qui semble naturel.

Comme presque tous les k-SIPs sont biologiquement cohérents, nous allons essayer de voir si une partie d'entre eux se détache des autres. En utilisant le coefficient de voisinage $\overline{w}_d$, nous évaluons la force du lien qui existe entre $\overline{w}_d$ et la p-valeur des k-SIPs. La Figure 4.1 permet de visualiser cela dans le cas de 1-$\mathcal{SIP}_{col}$. Le nuage de point ne semble pas indiquer l'existence d'un lien fort entre $\overline{w_d}$ et la p-valeur. Afin de confirmer ou infirmer cette observation, le coefficient de corrélation linéaire entre ces deux mesures, dans le cas de 1-$\mathcal{SIP}_{col}$ est calculé. Il est égal à $-0,272$, ce qui confirme bien que le lien entre $\overline{w_d}$ et la p-valeur est faible. Le calcul de la droite de régression linéaire, observable dans la Figure 4.1, ainsi que son R^2 qui est de 0,074, indiquent d'ailleurs qu'en fonction du 1-SIP, la force lien

Table 4.1 – Nombre moyen de gènes et de réactions (et écart-type) par k-SIP dans $\mathcal{G}_{col}$.

	1-SIP	2-SIP	3-SIP	4-SIP	5-SIP	6-SIP	7-SIP	8-SIP	9-SIP	10-SIP
# gènes	11,8	12,2	12,6	12,8	13,1	13,3	13,5	13,6	13,7	13,8
	$(\pm 4, 5)$	$(\pm 4, 5)$	$(\pm 4, 5)$	$(\pm 4, 6)$	$(\pm 4, 6)$	$(\pm 4, 6)$	$(\pm 4, 6)$	$(\pm 4, 7)$	$(\pm 4, 7)$	$(\pm 4, 7)$
# réactions	13,9	14,7	15,3	15,8	16,3	16,8	17,1	17,3	17,5	17,6
	$(\pm 5, 6)$	$(\pm 5, 6)$	$(\pm 5, 6)$	$(\pm 5, 7)$	$(\pm 5, 8)$	$(\pm 5, 9)$	$(\pm 6, 0)$	$(\pm 6, 1)$	$(\pm 6, 2)$	$(\pm 6, 2)$

entre $\overline{w_d}(sip)$ et la p-valeur de sip varie beaucoup. Ainsi, bien que l'information génomique des k-SIPs soit cohérente avec celle contenue dans la base de connaissance d'Ecocyc, elle ne suffit pas pour effectuer leur tri et mettre en évidence des k-SIPs intéressants.

4.5.2.2 Les k-SIPs de $\mathcal{G}_{col}$ passent par des gènes essentiels

Les gènes essentiels sont des gènes incontournables du génome. Nous allons étudier la façon dont ils apparaissent dans les k-SIPs. Tout d'abord, les 1-SIPs obtenus dans 1-$\mathcal{SIP}_{col}$ mettent en jeu 125 gènes essentiels, soit 93,98% des gènes essentiels métaboliques possibles. Dans le cas de 2-$\mathcal{SIP}_{col}$ et 3-$\mathcal{SIP}_{col}$, 130 (97,74%) gènes essentiels apparaissent, et 131 (98,50%) pour les 4-$\mathcal{SIP}_{col}$ à 10-$\mathcal{SIP}_{col}$.

Comme nous en retrouvons une quasi totalité, nous avons essayé de savoir si les gènes essentiels se différenciaient des gènes non essentiels dans l'ensemble k-$\mathcal{SIP}_{col}$, pour un k donné. Nous désignons par W_k l'ensemble des gènes qui apparaissent dans les k-SIPs de k-$\mathcal{SIP}_{col}$ et par W l'ensemble des gènes intervenant dans $\mathcal{G}_{col}$. Chaque gène g appartenant à W (ou W_k) est soit essentiel, soit non essentiel. Nous noterons l'ensemble des gènes essentiels intervenant dans $\mathcal{G}_{col}$ par Ess et l'ensemble des gènes non essentiels par $\overline{Ess}$ qui est complémentaire à Ess. Afin de tester si les gènes essentiels interviennent autant que les gènes non essentiels, nous avons étudié si la proportion de gènes essentiels qui interviennent dans k-$\mathcal{SIP}_{col}$ est similaire à la proportion de gènes essentiels présents dans $\mathcal{G}_{col}$. Nous utilisons la fonction différentielle suivante afin d'étudier ce rapport :

$$\Delta_{Ess}(W_k, W) = \frac{|\{g : g \in W_k, g \in Ess\}|}{|W_k|} - \frac{|\{g : g \in W, g \in Ess\}|}{|W|}$$

$\Delta_{Ess}(W_k, W)$ prend des valeurs comprises en -1 et 1. Pour un entier k donné, lorsque la valeur de $\Delta_{Ess}(W_k, W)$ est positive, cela signifie que les gènes essentiels sont proportionnellement plus présents dans W_k, l'ensemble des gènes intervenant dans k-$\mathcal{SIP}$, que dans W, l'ensemble des gènes intégrés à $\mathcal{G}_{col}$. Lorsque $\Delta_{Ess}(W_k, W)$ est négatif, alors c'est l'inverse : les gènes essentiels sont proportionnellement moins présents dans W_k que dans W. Les gènes non essentiels sont d'autant mieux mis en valeur que $\Delta_{Ess}(W_k, W)$ est petit. Quand $\Delta_{Ess}(W_k, W)$ est proche de 0, les gènes essentiels et non essentiels apparaissent, proportionnellement, autant dans k-$\mathcal{SIP}$ que dans $\mathcal{G}_{col}$.

La Figure 4.2 illustre l'évolution de $\Delta_{Ess}(W_k, W)$ en fonction de k. Pour $k = 1$, $\Delta_{Ess}(W_1, W)$ est très faible, mais les 1-SIPs tendent plus à passer par les gènes essentiels. Ce constat s'atténue au fur et à mesure que k augmente, car des chemins alternatifs sont ajoutés aux k-SIPs, ce qui accroît la diversité de gènes non essentiels intervenants dans k-$\mathcal{SIP}_{col}$. Les chemins passant par les gènes essentiels semblent donc le choix privilégié lorsqu'il existe plusieurs manières possibles pour aller d'une réaction à une autre dans $\mathcal{G}_{col}$.

Pour vérifier si de telles observations sont pertinentes, nous émettons l'hypothèse H_0 suivante que nous allons tenter de réfuter statistiquement : la distribution des gènes essentiels par rapport aux gènes non essentiels dans k-$\mathcal{SIP}_{col}$ est similaire à celle dans $\mathcal{G}_{col}$. Pour chaque k entre 1 et 10, nous avons réalisé un test du χ^2_1 entre chaque W_k et W pour les classes Ess et $\overline{Ess}$. Les résultats obtenus sont disponibles dans la Table B.1 de l'annexe B. Pour $k = 1$, le $\chi^2_1 = 0,542$ ne nous permet pas de réfuter H_0. C'est également le cas pour les autres valeurs de k. D'un point de vue statistique, que ce soit pour $k = 1$ ou $k = 10$, il n'est pas improbable d'obtenir un telle distribution des gènes essentiels et non essentiels dans W_k. Cependant, une partie des gènes non essentiels ne sont pas utilisés pour un petit k alors que nous calculons tous les chemins entre couples de réactions. Cela indique que certains gènes sont préférés à d'autres, mais à quel point ? Pour chaque valeur de k entre 1 et 10, l'étude de la fréquence

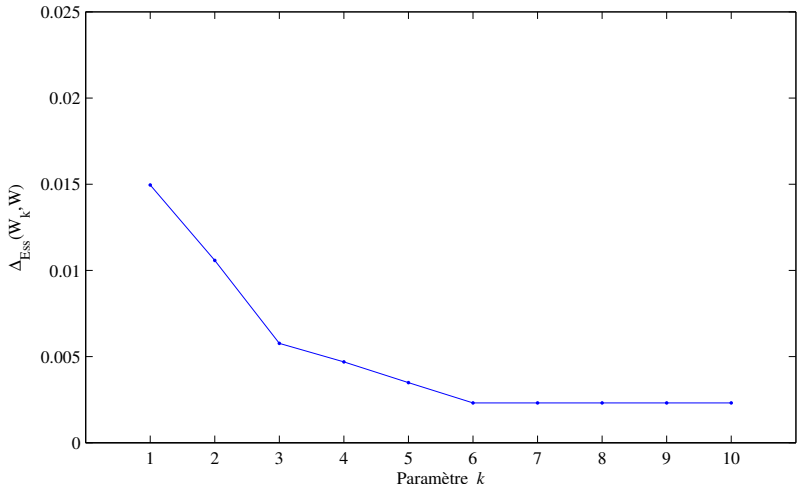


Figure 4.2 – Différence d'utilisation des gènes essentiels intervenant dans k-$\mathcal{SIP}_{col}$ et $\mathcal{G}_{col}$ en fonction de k. L'étude différentielle est réalisés en utilisant la formule $\Delta_{Ess}(W_k, W)$, avec W_k l'ensemble des gènes intervenant dans les k-SIPs de k-$\mathcal{SIP}_{col}$ et W l'ensemble des gènes intégrés à $\mathcal{G}_{col}$.

d'apparition d'un gène essentiel par rapport à un gène non essentiel au sein de k-$\mathcal{SIP}_{col}$ apparaît comme une piste à étudier pour démarquer les gènes essentiels des gènes non-essentiels.

Pour un k donné, un gène g qui intervient dans k-$\mathcal{SIP}_{col}$ apparaît dans un certain nombre de ces k-SIPs, que nous définissons comme le nombre d'occurrences de g. Nous utilisons le test bilatéral de Wilcoxon pour déterminer si la l'hypothèse H_1 est valide : dans k-$\mathcal{SIP}_{col}$, la fréquence d'occurrence moyenne des gènes essentiels est différente de la fréquence d'occurrence moyenne des gènes non-essentiels. Pour cela, nous essayons de rejeter statistiquement l'hypothèse nulle H_0 : les distributions des fréquences des gènes essentiels et non essentiels sont similaires. Dans le cas des 1-SIPs, nous avons obtenu une p-valeur de 0,5845. Cela signifie statistiquement que l'hypothèse nulle est non rejetable et donc que l'hypothèse H_1 n'est pas envisageable. Pour les autres k étudiés, nous obtenons des résultats similaires comme le montre la Table B.2 disponible dans l'annexe B.

Il est donc difficile de discriminer les gènes essentiels des gènes non-essentiels en étudiant leur fréquence d'apparition dans les k-SIP de $\mathcal{G}_{col}$. Il existe cependant d'autres entités biologiques connues sur lesquelles nous allons travailler.

4.5.2.3 Certains k-SIPs de $\mathcal{G}_{col}$ font apparaître des opérons

Dans cette partie, nous allons comparer chaque k-SIP obtenu (ou plutôt l'ensemble des gènes qui le compose) avec des groupes de gènes identifiés comme fonctionnellement liés : les opérons.

Étude relative au paramètre k

Dans un premier temps, nous nous intéressons à la variation du paramètre k, qui définit le nombre maximum de chemins alternatifs, dans l'identification de k-SIPs correspondant exactement, ou couvrant totalement les opérons (voir section 4.4.2 pour rappel). Pour chaque k entre 1 et 10, nous comparons chaque opéron avec chaque k-SIP et nous obtenons la Figure 4.3 où la courbe $\circ$ indique, pour chaque k, la proportion d'opérons couverts totalement par au moins un k-SIP. Nous avons également la courbe $\diamond$

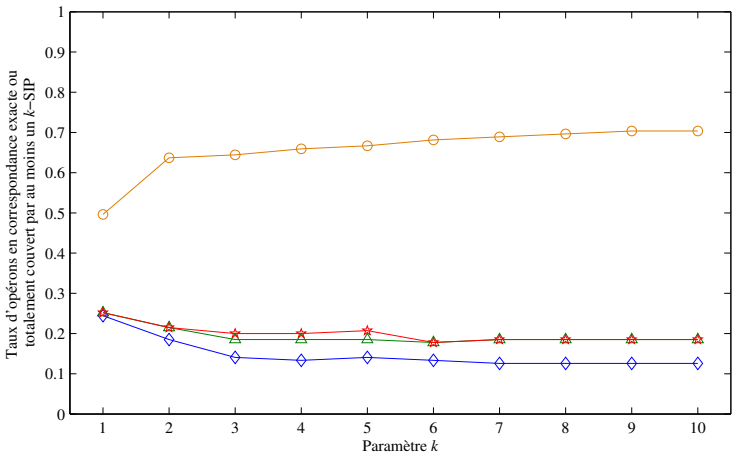


Figure 4.3 – Intérêt opéronique de tous les k-SIPs pour des valeurs de k distinctes dans $\mathcal{G}_{col}$. La courbe en $\circ$ représente les taux d'opérons qui sont couverts totalement par au moins un k-SIP. La courbe en $\diamond$ (les courbes en $\triangle$ et en $\star$ respectivement) présente le taux d'opérons seuls correspondant exactement à un k-SIP (le taux d'opérons seuls ou en couple correspondant exactement à un k-SIP, et le taux d'opérons seuls, en couple ou en triplet correspondant exactement à un k-SIP respectivement).

qui désigne la proportion d'opérons qui correspondent exactement à au moins un k-SIP (chacun de ces opérons produit, via tous ses gènes, les enzymes qui sont nécessaires pour catalyser les chaînes de réactions correspondantes), ainsi que les courbes $\triangle$ et $\star$ qui désignent respectivement la proportion de couples et de triplets d'opérons qui correspondent exactement à un unique k-SIP.

Nous observons que 24,4% des opérons correspondent exactement à un 1-SIP (soit 33 parmi 135, Table B.3 dans l'annexe B pour la liste complète), et que 49,73% des opérons sont couverts totalement par au moins un 1-SIP (courbe $\circ$ dans la Figure 4.3). Le taux d'opérons correspondant exactement à un 10-SIPs tombe à 12,59% et le taux d'opérons couvert totalement par un 10-SIP monte à 64,44%. Au fur et à mesure que k augmente, les k-SIPs contiennent davantage de chemins, et tendent alors à former des ensembles de gènes qui incluent des opérons, comme le montre la croissance de la courbe $\circ$ dans la Figure 4.3, tandis que le taux de k-SIPs correspondant exactement à un opéron diminue, comme le montre la décroissance de la courbe $\diamond$ dans la Figure 4.3. Cela indique que les k-SIPs tendent à être un peu plus que des opérons. Notons que 8,15% des opérons (11 d'entres eux) correspondent exactement à un k-SIP pour toutes les valeurs de k de 1 à 10. C'est principalement dû au fait que dans $\mathcal{G}_{col}$ il n'existe pas d'autre chemin que celui du 1-SIP. Le 10-SIP est alors identique au 1-SIP et indique que ces k-SIPs contiennent probablement des 'choke-points' du réseau métabolique, décrits à la section 2.2.2 [82]. Au final, pour k entre 1 et 10, 37,03% des opérons (soit 50 opérons) ont une correspondance exacte avec des k-SIPs.

Nous répétons cette comparaison entre chaque k-SIP et, cette fois-ci, chaque couple (et triplet respectivement) d'opérons. Nous trouvons 14 couples (2 triplets respectivement) d'opérons qui correspondent exactement à un k-SIP, pour différentes valeurs de $k \geq 1$ (voir dans la Table B.4 dans l'annexe B pour plus de détails). Dans la Figure 4.3, la ligne $\triangle$ (la ligne $\star$ respectivement) montre que le taux d'opérons en exacte correspondance avec un k-SIP croît quand les opérons seuls et couples d'opérons (opérons seuls, couples et triplets d'opérons respectivement) sont considérés. Aucun résultat n'a été trouvé pour

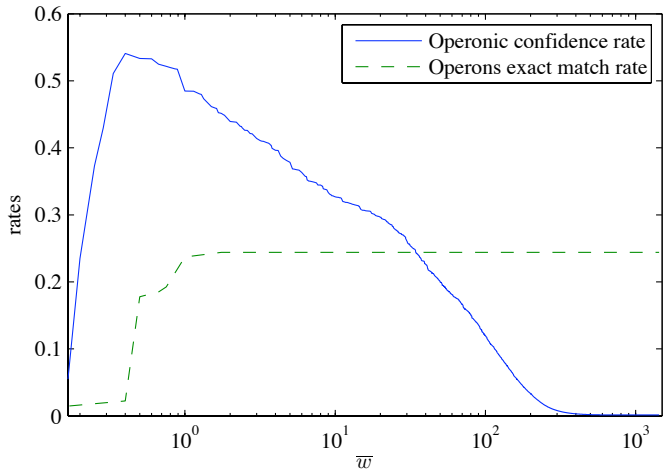


Figure 4.4 – Taux de 1-SIPs qui correspondent exactement à un opéron en fonction de $\overline{w}_d$ dans $\mathcal{G}_{col}$. La ligne pleine montre l'évolution de ce taux quand $\overline{w}_d$ est borné par un seuil supérieur marqué sur l'axe des abscisses (échelle logarithmique). En complément, la ligne discontinue représente le taux d'opérons correspondant exactement à un 1-SIP avec un $\overline{w}_d$ borné par cette valeur donnée.

des correspondances de 4-uplets d'opérons ou plus avec un k-SIP. Ainsi, notre méthode `SIPPER` permet de décrire non seulement un opéron seul par k-SIP, mais aussi des couples et des triplets d'opérons contrairement aux autres méthodes. L'analyse ontologique dans la base de connaissance d'Ecocyc avec `GO::TermFinder`, disponible dans la Table B.4 en annexe B, confirme d'ailleurs l'intérêt fonctionnel des opérons à être considérés ensemble plutôt que seuls. En effet, nous observons un gain d'information indiqué par une p-valeur plus faible quand les opérons sont regroupés.

Étude relative au coefficient de voisinage $\overline{w}_d$

Nous étudions ensuite l'évolution du taux de k-SIPs qui correspondent exactement à un opéron chacun, en fonction de la valeur $\overline{w}_d$ du k-SIP. Nous appelons ce taux le *taux de confiance opéronique*. Pour plus de clarté dans notre propos, nous nous intéressons à l'ensemble 1-$\mathcal{SIP}_{col}$ qui présente le plus grand taux d'opérons correspondant exactement à un 1-SIP chacun (voir Figure 4.3). La Figure 4.4 détaille l'évolution du taux confiance opéronique des 1-SIPs (ligne continue) quand $\overline{w}_d$ est borné de façon supérieure par la valeur sur l'axe des abscisses. Ce taux augmente pour $\overline{w}_d$ entre 0 et 0,5 en faisant apparaître deux faits majeurs. Premièrement, les 1-SIPs qui correspondent à des opérons contiennent des réactions successives catalysées par un même gène. En effet, étant donnée la définition de la mesure $\overline{w}_d$, elle ne peut être plus petite que $0,5$ que s'il existe plusieurs occurrences successives d'un même gène dans un k-SIP. Les 1-SIPs correspondants aux opérons *fadIJ* et *fadBA* dans la Table B.3 en sont des exemples. Deuxièmement, l'ordre des gènes dans le génome et celui des réactions qu'ils catalysent est similaire : chaque arc contribue généralement de 1 dans le poids du 1-SIP (sauf quand il y a plusieurs occurrences successives d'un même gène), deux réactions successives étant catalysées par le produits de deux gènes contigus. Au delà de 0,5, l'augmentation de $\overline{w}_d$ conduit à une lente diminution du taux de confiance opéronique. Le taux de confiance opéronique dépasse 0,5 pour une valeur de $\overline{w}_d$ entre 0,4 et 1. La courbe illustrée par la ligne discontinue présente le taux d'opérons correspondant exactement à un 1-SIP. Ce taux

Table 4.2 – Résumé des correspondances exactes entre les k-SIPs ($k = 1$ et $k = 10$) et les opérons d'*E. coli* en fonction de la valeur de $\overline{w}_d$ dans $\mathcal{G}_{col}$.

Ensemble de données	k	Taux d'opérons correspondant exactement aux k-SIPs		
		$\overline{w}_d \leq 1.0$	$\overline{w}_d \leq 5.0$	$\overline{w}_d \leq 200.0$
k-$\mathcal{SIP}_{col}$	1	23.71%	24.44%	24.44%
	10	8.15%	12.59%	12.59%
k-$\mathcal{SIP}_{col}$ avec un génome aléatoire	1	0%	0%	2.22%
	10	0%	0%	0.74%
k-$\mathcal{SIP}_{col}$ avec un réseau métabolique aléatoire	1	1.48%	1.48%	1.48%
	10	0%	0%	0%

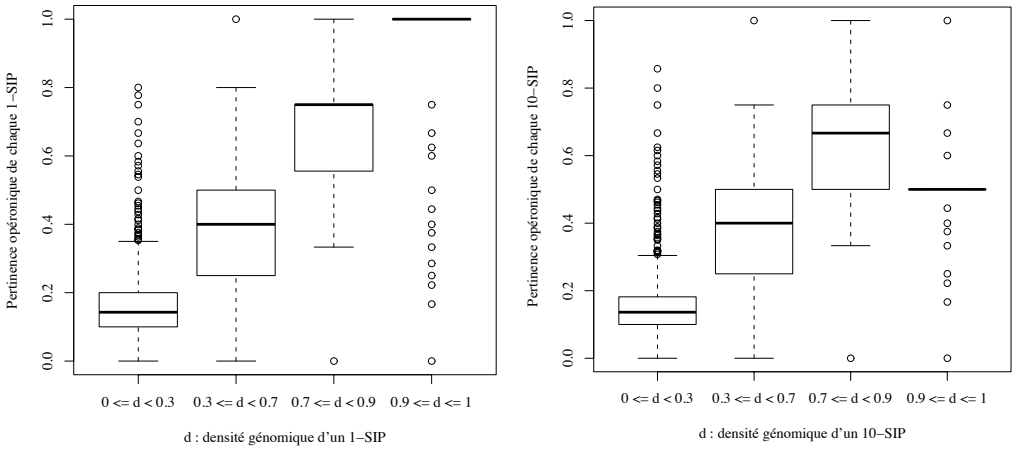
croît rapidement entre $\overline{w}_d = 0,4$ et $0,5$, puis croît plus lentement jusqu'à $\overline{w}_d = 1$ pour enfin atteindre un plateau à $\overline{w}_d$ de 1,75, ce qui signifie qu'au delà de $\overline{w}_d = 1,75$ aucun nouvel opéron n'est plus reconnu. Ces résultats indiquent que les k-SIPs "opéroniques" ont un petit $\overline{w}_d$. Quand $\overline{w}_d \leq 1$ plus de la moitié des 1-SIPs correspond exactement à 23,71% des opérons. Ce résultat constitue un seuil envisageable pour $\overline{w}_d$ afin de discriminer les 1-SIPs opéroniques.

La Table 4.2 reprend en partie les résultats précédents en comparant entre eux les taux d'opérons correspondant exactement à un k-SIP (pour $k = 1$ et 10) dans *E. coli* pour les données aléatoires ou non. L'ordre aléatoire des gènes (ligne k-$\mathcal{SIP}_{col}$ avec génome aléatoire) change significativement la distance de colocalisation entre les gènes d'un même opéron d'*E. coli*. Cela entraîne une augmentation importante de $\overline{w}_d$ pour les 1-SIPs qui correspondent exactement à un opéron. Ces 1-SIPs existent toujours car le réseau métabolique n'a pas changé, mais il n'est plus possible de les identifier à l'aide de $\overline{w}_d$. Un mélange du réseau métabolique quant à lui (ligne k-$\mathcal{SIP}_{col}$ avec réseau métabolique aléatoire) change beaucoup les chemins à l'intérieur du réseau d'*E. coli*. Il en résulte qu'un petit nombre d'opérons est retrouvé par chance.

Pour résumer, la mesure $\overline{w}_d$, quand elle est petite, permet de mettre en évidence des opérons métaboliques (plus de 50% des 1-SIPs avec un $\overline{w} \leq 1$ sont des opérons exacts) qui ne sont pas dus au hasard. Cette conclusion n'est valable ici que dans $\mathcal{G}_{col}$. Ces résultats encourageants nous indiquent que minimiser $\overline{w}_d$ dans $\mathcal{G}_{int}$, pondéré par une distance de colocation, semble être une bonne mesure dans la prédiction d'opérons.

Étude relative à la densité génomique

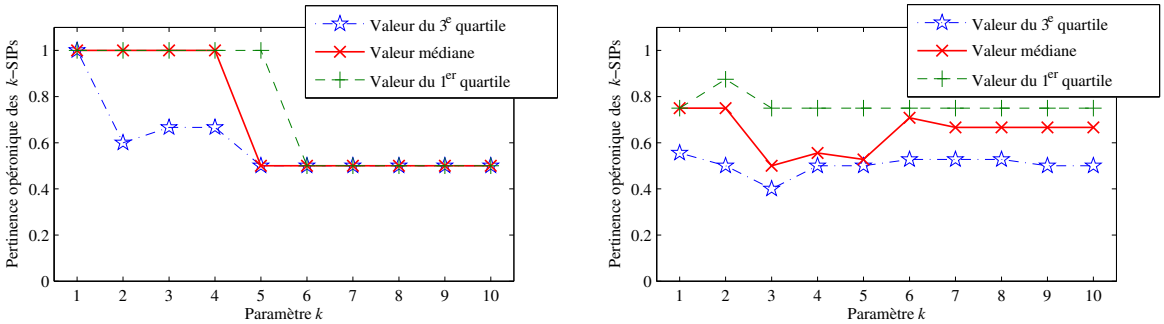
Lorsque la mesure $\overline{w}_d$ d'un k-SIP est inférieure à 1, les gènes mis en jeu sont proches sur le génome, mais pas forcément successifs. La densité génomique d_G, définie dans la section 3.3.2 à la page 29, est une autre mesure pour discriminer les chemins qui contiennent uniquement des gènes contigus. Elle nous semble une mesure intéressante à étudier dans la sélection de k-SIPs correspondant à des opérons. La Figure 4.5 illustre, pour les ensembles 1-$\mathcal{SIP}_{col}$ et 10-$\mathcal{SIP}_{col}$, la relation qui existe entre la densité génomique des k-SIPs et leur pertinence opéronique définie dans la section 4.4.2. Nous observons une augmentation de la pertinence opéronique au fur et à mesure que la densité génomique augmente, en particulier pour $k = 1$. Ceci est moins vrai pour $k = 10$, mais les 10-SIPs qui ont une densité génomique comprise entre 0,7 et 0,9 sont suffisamment pertinents pour prédire des opérons. La Figure 4.6 montre plus en détail l'évolution des différentes classes de densité génomique des k-SIPs en fonction du para-



(a) (b)

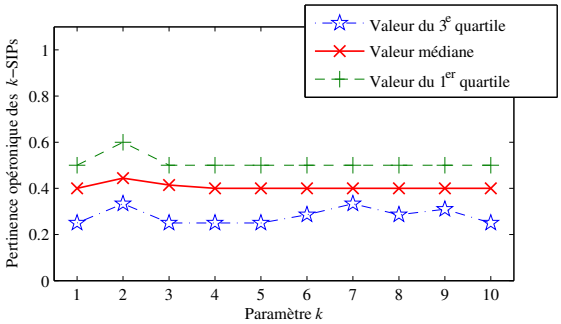
Figure 4.5 – Taux de correspondance, pour $k = 1$ et 10, entre les k-SIPs de k-$\mathcal{SIP}_{col}$ regroupés par densité génomique et les opérons d'*E. coli* en utilisant la mesure de Jaccard. Pour chaque k-SIP sip, la densité génomique $d_G(sip)$ et la pertinence opéronique $P_o = \max_{op \in \textit{Opéron}}\{\textit{Jaccard}(sip, op)\}$ donnent les abscisses et ordonnées d'un point dans le graphe. Les points sont regroupés en quatre classes de densité (axe des abscisses), et une boîte est dessinée dans chacune de ces classes afin de résumer, pour les k-SIPs de la classe donnée, le score P_o (axe des ordonnées). Les limites supérieures et inférieures d'une boîte sont les valeurs des 1^{er} et 3^e quartiles de la classe qu'elle résume, et chaque extension au delà des limites de la boîte couvre les k-SIPs qui ne sont pas plus distant de 1,5 fois la hauteur de la boîte. Les points en dehors de ces extensions correspondent aux k-SIPs qui se différencient trop de ceux dans la boîte.

mètre k. À partir de $k > 4$, les k-SIPs de densité supérieure à 0,9 sont moins pertinents pour décrire des opérons métaboliques (la mesure de Jaccard entre les k-SIPs et les opérons passe de 1 à 0,5) alors que ceux qui ont une densité comprise entre 0,7 et 0,9 sont toujours assez pertinents (la mesure opéronique des k-SIPs est comprise entre 0.7 et 0.5). Ceci est dû au fait que lorsque k augmente, il y a plus de gènes qui s'ajoutent aux opérons identifiés quand k est plus petit. Cet enrichissement peut sembler gênant pour prédire des opérons, mais il se révèle intéressant, car il correspond à des gènes qui agissent comme des acteurs supplémentaires aux gènes opéroniques. Cela souligne une dépendance fonctionnelle potentielle entre ce(s) gène(s) et l'opéron. Ce résultat confirme l'intérêt de faire varier le paramètre k, et donc de ne pas calculer que des 1-SIPs dans l'approche `SIPPER`. Ce résultat souligne aussi le fait que si la mesure $\overline{w}_d$ discrimine des opérons surtout dans le cas $k = 1$, la densité génomique, lorsque $d_G \geq 0,7$ les discrimine quelle que soit la valeur de k.

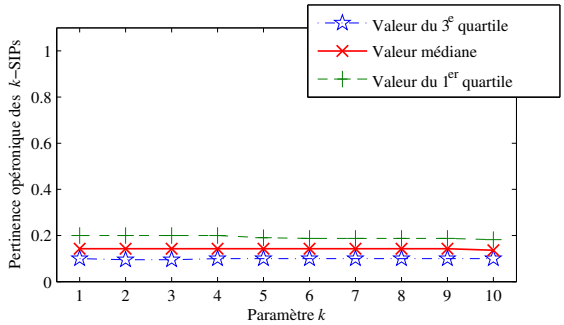


(a) $0.9 \leq d_G \leq 1.0$

(b) $0.7 \leq d_G < 0.9$



(c) $0.3 \leq d_G < 0.7$



(d) $0.0 \leq d_G < 0.3$

Figure 4.6 – Évolution de la pertinence opéronique des k-SIPs de $\mathcal{G}_{col}$ groupés par classe de densité génomique pour $1 \leq k \leq 10$. L'évolution de chaque classe de densité génomique : (a) $0.9 \leq d_G \leq 1.0$, (b) $0.7 \leq d_G < 0.9$, (c) $0.3 \leq d_G < 0.7$ and (d) $0.0 \leq d_G < 0.3$, est résumée par trois courbes dans chaque cas : les courbes $\times$, $+$ et $\star$ représentent respectivement l'évolution de la valeur médiane, la valeur du 1^{er} et 3^e quartile de pertinence opéronique des k-SIPs de densité d.

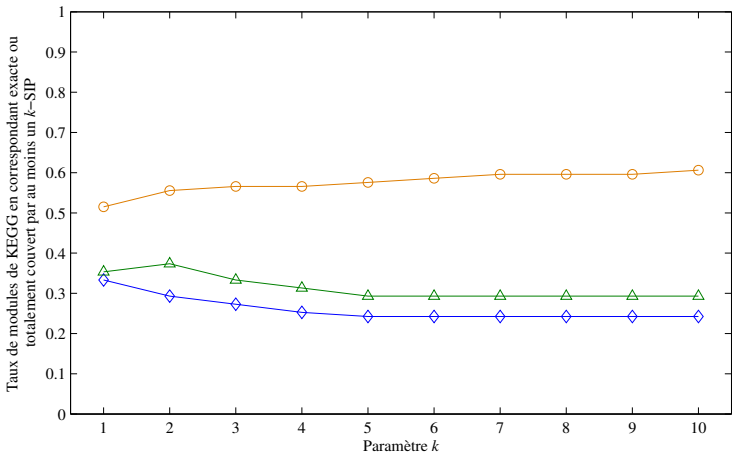


Figure 4.7 – Intérêt modulaire de tous les k-SIPs de k-$\mathcal{SIP}_{col}$ pour des k distincts dans *E. coli*. La courbe $\circ$ représente les taux de modules de KEGG qui sont couverts totalement par au moins un k-SIP. Le courbe $\diamond$ (la courbe $\triangle$ respectivement) résume le taux de modules seul (ou en couple respectivement) qui correspondent exactement à des k-SIPs.

4.5.2.4 Des k-SIPs de $\mathcal{G}_{col}$ correspondent à des modules métaboliques

De la même manière dont nous avons comparé des k-SIPs avec les opérons, nous avons comparé les k-SIPs obtenus avec les modules métaboliques de KEGG.

Étude relative au paramètre k

Tout d'abord, nous étudions le rôle de la variation du paramètre k dans l'identification des modules de KEGG. La Figure 4.7 résume les résultats obtenus tant au niveau de la couverture totale des modules par les k-SIPs qu'au niveau de la correspondance exacte entre modules et k-SIPs (rappels disponibles à la section 4.4.2). Nous avons trouvé une correspondance exacte entre 33,33% des modules et les 1-SIPs (soit 33 modules parmi 99, voir ligne $\triangle$ dans la Figure 4.7 et la Table B.5 dans l'annexe B pour une liste précise). Nous constatons également que 52,52% des modules sont couverts totalement par au moins un 1-SIP (ligne $\circ$ dans la Figure 4.7). Ces deux taux tombent et montent respectivement à 26,26% et 60,60% quand $k = 10$. Nous observons aussi que 18,18% des modules de KEGG ont toujours une correspondance exacte pour toutes les valeurs de k entre 1 et 10.

D'un point de vue qualitatif, lorsque nous avons comparé les couples de modules et les k-SIPs, nous avons trouvé que 16 couples de modules correspondent exactement à des k-SIPs pour $k \geq 1$ (voir la Table B.6 et la Table B.7 en annexe B pour plus de détails). Dans la Figure 4.7, la ligne $\triangle$ montre l'augmentation du taux de modules identifiés quand les couples de modules sont considérés en plus des modules seuls. Aucun n-uplet de modules plus grand qu'un couple n'a été identifié parmi les k-SIPs. Nous avons donc des k-SIPS qui correspondent chacun à des modules seuls ou à des couples de modules des KEGG. Il s'agit maintenant de les différencier des autres k-SIPs à l'aide de $\overline{w}_d$ ou d_G.

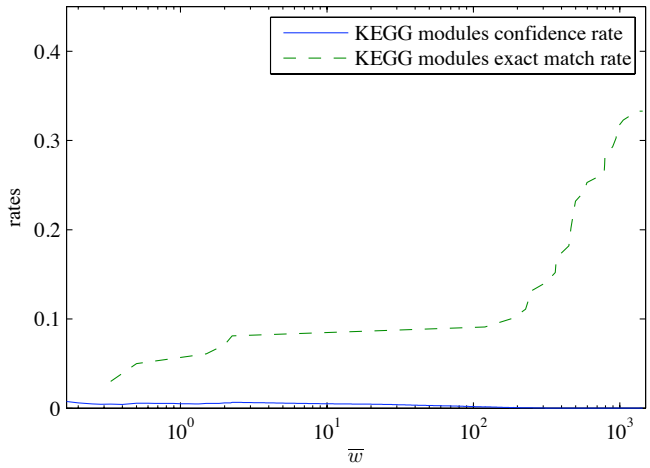


Figure 4.8 – Taux de 1-SIPs de 1-$\mathcal{SIP}_{col}$ qui correspondent exactement à un module de KEGG en fonction de $\overline{w}_d$ dans *E. coli*. La courbe pleine montre l'évolution de ce taux quand $\overline{w}_d$ est borné de façon supérieure par une valeur donnée sur l'axe des abscisses. La courbe discontinue représente le taux de modules en correspondance exacte avec un 1-SIP pour la borne supérieure de $\overline{w}_d$ donnée.

Étude relative au coefficient de voisinage $\overline{w}_d$

La Figure 4.8 résume une partie des résultats obtenus à propos de $\overline{w}_d$. Pour les 1-SIPs, nous y observons, en fonction d'une borne maximum sur $\overline{w}_d$ (axe des abscisses), l'évolution de deux taux (axe des ordonnées) : le *taux de confiance modulaire*, qui est la proportion de 1-SIPs qui correspondent exactement à un module de KEGG, et le taux de modules en correspondance exacte avec un 1-SIP. L'étude de cette figure ne fait pas apparaître de valeur particulière de $\overline{w}_d$ qui met en évidence les k-SIPs correspondant exactement à des modules de KEGG. En effet, le taux de confiance modulaire reste extrêmement bas quelle que soit la valeur de $\overline{w}_d$. Il n'existe pas non plus de valeur de $\overline{w}_d$ pour laquelle nous observons une forte augmentation du nombre de 1-SIPs en correspondance exacte avec des modules. Cette observation est d'ailleurs confirmée par la liste des k-SIPs de la Table B.5 dans l'annexe B. Toutefois, la Table 4.3 indique que certains modules de KEGG sont constitués de gènes proches sur le génome. Cette information est d'ailleurs confirmée lorsque nous regardons la constitution des k-SIPs de la Table B.7 dans l'annexe B.

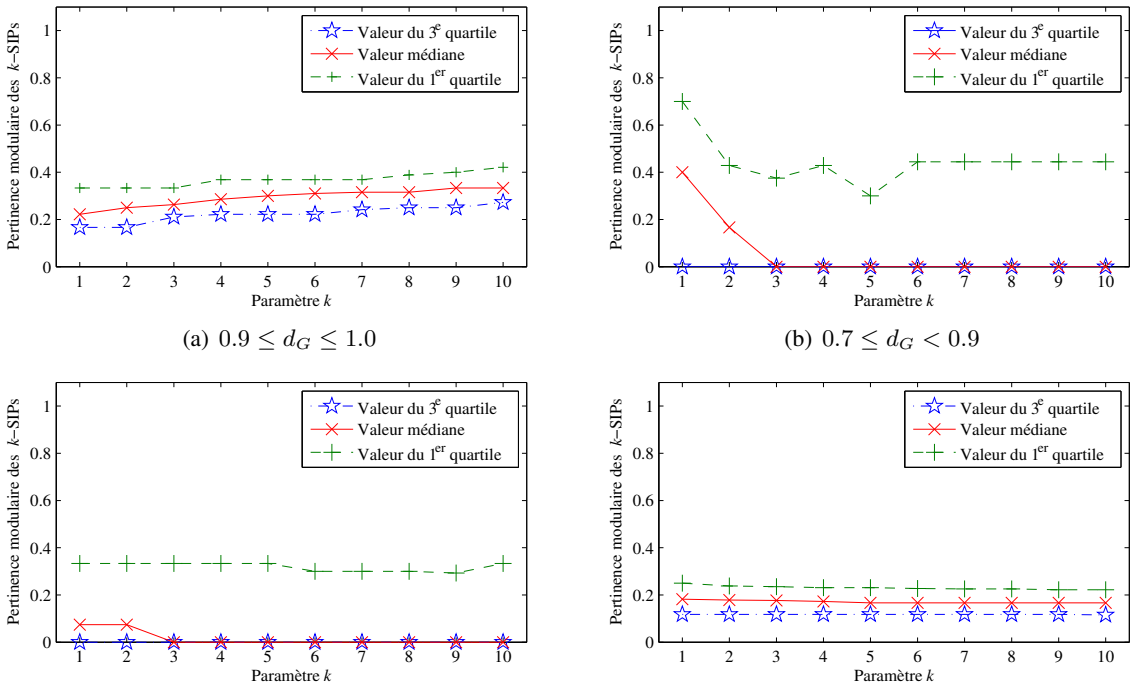
Étude relative à la densité génomique

L'étude de la pertinence modulaire des k-SIPs (voir section 4.4.2) en fonction et de la densité génomique et de k est résumée dans la Figure 4.9. Nous y observons que très peu de modules de KEGG sont formés de groupes de gènes denses. En effet, les k-SIPs de densité supérieure à $0,7$ pour $k = 1$ et entre $0,9$ et $1,0$ pour $2 \leq k \leq 10$ ont une pertinence modulaire légèrement meilleure que celle des k-SIPs affichant une densité plus faible (inférieure à 0,7).

Cette constatation, ainsi que les résultats observés lors de l'étude de $\overline{w}_d$, montrent que les modules de KEGG sont en partie définis par des gènes proches sur le génome, ce qui est en accord avec leur définition. En étudiant la Figure 4.10, nous constatons que 20% des 1-SIPs couvrent totalement un ou plusieurs opérons uniquement et que 10% d'entre-eux couvrent totalement un ou plusieurs opérons mo-

Table 4.3 – Résumé des correspondances exactes entre les k-SIPs ($k = 1$ et $k = 10$) et les modules de KEGG pour *E. coli* en fonction de la valeur de $\overline{w}_d$.

Ensemble de données	k	Taux de modules de KEGG en correspondance exacte avec un k-SIP		
		$\overline{w} \leq 1.0$	$\overline{w} \leq 5.0$	$\overline{w} \leq 200.0$
k-$\mathcal{SIP}$	1	5.05%	8.08%	10.10%
	10	1.01%	3.03%	3.03%
k-$\mathcal{SIP}$ avec un génome aléatoire	1	2.02%	3.03%	6.06%
	10	0%	0%	0%
k-$\mathcal{SIP}$ avec un réseau métabolique aléatoire	1	0%	0%	0%
	10	0%	0%	0%



(a) $0.9 \leq d_G \leq 1.0$ (b) $0.7 \leq d_G < 0.9$

(c) $0.3 \leq d_G < 0.7$ (d) $0.0 \leq d_G < 0.3$

Figure 4.9 – Évolution de la pertinence modulaire des k-SIPs de $\mathcal{G}_{col}$ groupés par classe de densité génomique et pour $1 \leq k \leq 10$. L'évolution de chaque classe de densité génomique : (a) $0.9 \leq d_G \leq 1.0$, (b) $0.7 \leq d_G < 0.9$, (c) $0.3 \leq d_G < 0.7$ et (d) $0.0 \leq d_G < 0.3$, est résumée par trois courbes dans chaque cas : les courbes $\times$, $+$ et $\star$ représentent respectivement l'évolution de la valeur médiane, la valeur du 1^{er} et du 3^e quartile de la pertinence modulaire des k-SIPs de densité d.

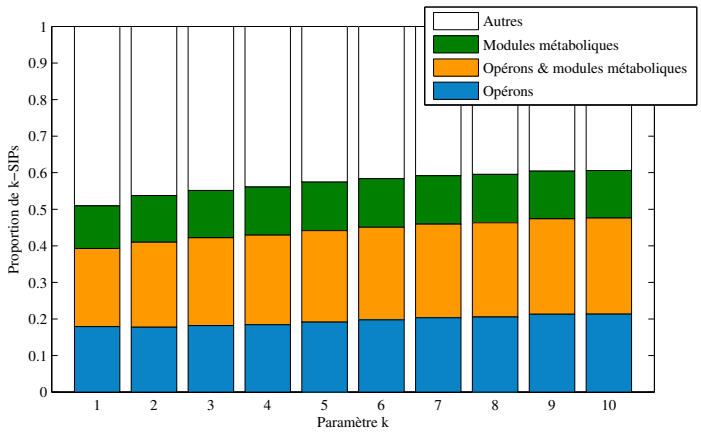


Figure 4.10 – Les différentes familles de k-SIPs possibles. Un k-SIP couvrant totalement : un ou plusieurs opérons appartient à "Opérons", un ou plusieurs modules de KEGG appartient à "Modules métaboliques", un ou plusieurs opérons et un ou plusieurs modules de KEGG appartient à "Opérons & modules métaboliques". Les k-SIPs n'appartenant pas à ces trois familles appartiennent à la famille "Autres". Pour chaque k entre 1 et 10 (abscisses), l'ensemble des k-SIPs est réparti dans ces différentes familles. La proportion de k-SIPs appartenant à chaque famille est indiquée sur l'axe des ordonnées.

dules métaboliques uniquement. Nous constatons également que 20% des 1-SIPs couvrent totalement et simultanément un ou plusieurs modules métaboliques et un ou plusieurs opérons. Cela indique qu'il existe un lien topologique entre opérons et modules. Cette proportion croît au fur et à mesure que k augmente. Si une partie des k-SIPs couvrent totalement et simultanément des opérons et des modules, les k-SIPs en correspondance exacte avec des opérons ou modules de KEGG sont cependant distincts. Seul un 1-SIP correspond exactement à la fois à un opéron (l'opéron *fabBA*) et à un module de KEGG (le module *M00059*), ce qui indique que les recherches d'opérons et de modules de KEGG ne dépendent pas des mêmes informations. Nous avons montré que la densité génomique permet de discriminer les k-SIPs qui sont des opérons, mais nous n'avons pas de critère pour distinguer les modules de KEGG. Comme, par définition, un module est constitué manuellement à partir de résultats venant de différentes méthodes, la distance de colocalisation utilisée dans $\mathcal{G}_{col}$ ne convient pas. L'utilisation d'une pondération de $\mathcal{G}_{int}$ constituée du cumul de plusieurs distances, dont celle de colocalisation, chacune encodant une des notions prises en compte dans la constitution d'un module apparaît comme une solution adaptée pour retrouver des modules métaboliques.

4.5.3 Comparaison avec d'autres approches

Comme au chapitre précédent, nous avons comparé notre méthode avec d'autres approches, nous allons maintenant comparer les résultats que nous avons obtenus à ceux obtenus par ces autres approches. Nous allons d'abord comparer les k-SIPs aux FRECs (clusters d'enzymes liés fonctionnellement, section 2.4.2) [78], puis aux CCCs (composantes connexes communes, section 2.4.2) [20] et, dans le cadre de la prédiction d'opérons, nous allons également comparer notre approche à celle de Zheng *et al.* [110].

Comparaison avec les FRECs

Nous avons essayé de voir si nous retrouvions des résultats similaires à ceux obtenus par Ogata *et al.* [78]. Les auteurs ne fournissent pas de liste des FRECs qu'ils ont obtenus, mais nous avons tout de même vérifié si nous retrouvons l'exemple proposé dans leur papier. La Figure 4.11 est un exemple de FREC retrouvé par Ogata *et al.* Il est possible de le retrouver indirectement en faisant l'union du 1-SIP allant de r_1 à r_6 et du 1-SIP allant de r_7 à r_6, ou directement, en recherchant le 2-SIP allant de $R_1 = \{r_1, r_7\}$ à $R_2 = \{r_6\}$.

Comparaison avec les CCCs

Lors du calcul de CCCs [20] entre le génome et le métabolisme, il faut fixer les paramètres δ_g et δ_r lors de la création du multi-graphe. Le paramètre δ_g désigne la distance seuil maximale, en nombre de gènes, jusqu'à laquelle deux gènes sont liés par une arrête dans le multi-graphe. De même, le paramètre δ_r désigne la distance seuil maximale sur le réseau métabolique, en nombre de réactions, jusqu'à laquelle deux réactions sont liées par une arrête dans le multi-graphe.

La Table 4.4 présente les résultats obtenus dans l'identification d'opérons métaboliques par les CCCs, d'une part, et par les k-SIPs en intégrant le réseau métabolique et le génome d'*E. coli*, d'autre part. Les données utilisées pour cette comparaison sont celles intégrées à $\mathcal{G}_{col}$. Nous avons utilisé, pour les CCCs, les valeurs de (δ_g, δ_r) présentées dans [20] : (0,0), (1,0), (0,1) et (5,3). Nous avons comparé les CCCs aux k-$\mathcal{SIP}$s avec $k = 1, 2, 3$ et 10.

Nous observons que les k-SIPs ont un taux de couverture totale (colonne (a)) ou demi-couverture (colonne (c)) d'un opéron par un k-SIP plus important que celui des CCCs pour des valeurs de k, δ_g et δ_r faible. Cette différence s'amoindrit pour des valeurs de k, δ_g et δ_r plus importantes, soulignant que l'ajout de chemins alternatifs dans les k-SIPs, comme l'interconnexion de gènes et de réactions de plus en plus distants dans les CCCs, permet de couvrir une grande partie des opérons. L'analyse du taux de correspondance exacte d'un opéron et d'un k-SIP (colonne (b)) fait apparaître que les CCCs permettent d'identifier plus d'opérons. Cela est dû principalement au fait que les CCCs permettent de lier des éléments distants en omettant des gènes ou réactions intermédiaires et forment des sous-graphes non orientés, alors que les k-SIPs forment un ensemble de chemins orientés. Cette variation de résultat est d'ailleurs confirmée lorsque nous nous intéressons à la demi couverture mutuelle (colonne (d)), où le taux de reconnaissance des k-SIPs est similaire à celui des CCCs : la tolérance d'erreur induite par le seuil de 50% au niveau de la mesure de Jaccard rapproche les résultats obtenus par les deux méthodes.

Les approches présentées dans [78, 20] introduisent des paramètres de plasticité pour prendre en compte les informations manquantes. Au contraire des autres approches, la plasticité de SIPPER est inhérente à la méthode et s'appuie sur le paramètre k.

Prédiction d'opérons

Nous avons comparé notre approche `SIPPER` avec des méthodes automatiques de prédiction d'opérons. En considérant tous les k-SIPs de k-$\mathcal{SIP}_{col}$ pour une valeur de k donnée, nous obtenons la Table 4.5. La correspondance exacte d'opérons avec les k-SIPs décroît quand k augmente, mais les mesures plus traditionnelles de sensibilité (qui désigne le taux de paires de gènes successifs à l'intérieur d'un opéron correctement prédites) et de spécificité (qui désigne le taux de paires de gènes en bordure d'unité de transcription correctement prédites) s'améliorent clairement jusqu'à $k = 6$, puis reculent un peu et restent stables pour $k > 6$. Nos résultats sont comparables à ceux présentés dans [110] (89% de sensibilité et 87% de spécificité), qui portent plus particulièrement sur les opérons métaboliques.

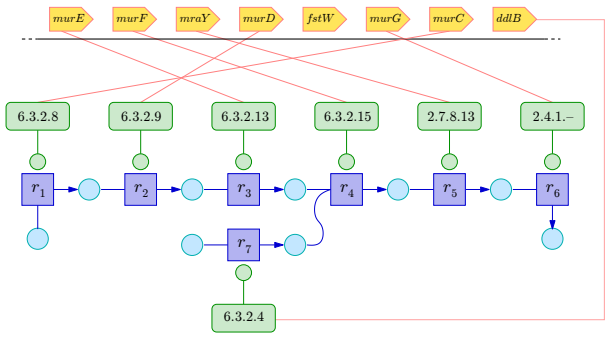


Figure 4.11 – Sept enzymes, et leur gènes codants, associées à la biosynthèse du peptidoglycan dans *E. coli*.

Table 4.4 – Comparaison des k-SIPs et des CCCs dans l'identification des opérons métaboliques d'*E. coli*. Cette table à double entrée présente le taux d'opérons dans *E. coli* dont l'ensemble de gènes (a) est couvert totalement (colonne Couverture = 1), respectivement (c) couvert au moins à moitié (colonne Couverture $\geq$ 0,5) par un k-SIP et (b) est en correspondance exacte (colonne Mesure de Jaccard =1), respectivement (d) est au moins en demi-correspondance (colonne Mesure de Jaccard $\geq$ 0,5) avec un k-SIP. L'expérience a été réalisée sur l'ensemble du métabolisme d'*E. coli*. Différentes valeurs de k, δ_g et δ_r on été choisies pour autoriser des sauts à la fois dans le génome et le réseau métabolique.

		(a) Couverture = 1	(b) Mesure de Jaccard = 1	(c) Couverture $\geq$ 0,5	(d) Mesure de Jaccard $\geq$ 0,5
k-SIPs	1-$\mathcal{SIP}_{col}$	49.63%	24.44%	82.96%	56.30%
	2-$\mathcal{SIP}_{col}$	63.70%	18.52%	87.41%	61.48%
	3-$\mathcal{SIP}_{col}$	64.44%	14.07%	91.11%	59.26%
	10-$\mathcal{SIP}_{col}$	70.37%	12.59%	92.59%	52.59%
CCC(δ_g, δ_r)	CCC$(0,0)$	38.93%	36.64%	52.67%	50.38%
	CCC$(1,0)$	47.33%	39.69%	60.31%	58.02%
	CCC$(0,1)$	47.33%	43.51%	67.18%	63.36%
	CCC$(5,3)$	71.76%	32.82%	89.31%	60.31%

Table 4.5 – Taux d'opérons identifiés dans chaque k-$\mathcal{SIP}_{col}$. Le taux de correspondance exacte est calculé en utilisant la mesure de Jaccard comme mesure de similarité. Les résultats sont aussi décrits en utilisant les mesures standard de prédiction d'opérons [91] : la sensibilité désigne le taux de paires de gènes consécutifs à l'intérieur d'un opéron (within operon gene pairs) qui sont correctement prédites, et la spécificité désigne le taux de paires de gènes en bordure d'unité de transcription (transcriptional unit border genes pairs) qui sont correctement prédites.

	1-$\mathcal{SIP}_{col}$	2-$\mathcal{SIP}_{col}$	3-$\mathcal{SIP}_{col}$	4-$\mathcal{SIP}_{col}$	5-$\mathcal{SIP}_{col}$	6-$\mathcal{SIP}_{col}$	7-$\mathcal{SIP}_{col}$	8-$\mathcal{SIP}_{col}$	9-$\mathcal{SIP}_{col}$	10-$\mathcal{SIP}_{col}$	Zheng *et al.*
Correspondance exacte	24.44%	18.52%	14.07%	13.33%	14.07%	13.33%	12.59%	12.59%	12.59%	12.59%	-
Sensibilité	61.93%	76.14%	79.69%	81.21%	82.23%	83.75%	83.75%	83.75%	83.75%	83.75%	89%
Spécificité	77.26%	85.75%	87.67%	89.31%	89.86%	90.13%	89.86%	89.86%	89.86%	89.86%	87%

4.5.4 Résumé

L'application de SIPPER sur $\mathcal{G}_{col}$ ne nous permet pas de mettre en valeur les gènes essentiels, mais en nous intéressant aux mesures de coefficient de voisinage $\overline{w}_d$ et de densité génomique d_G, nous avons mis en évidence des opérons métaboliques. SIPPER, contrairement à d'autres approches comme [110, 27], n'a pas pour vocation l'identification d'opérons, mais réalise l'analyse précise du lien entre une hypothèse biologique, via une distance d, et des enchaînements de réactions. La distance de colocalisation, que nous avons utilisée, est une mesure courante dans la littérature, mais elle est rarement évaluée de façon automatique comme nous l'avons fait [79, 12]. La variation du paramètre k nous permet d'identifier plusieurs opérons. Cette variation met également en évidence que l'opéron seul n'est pas forcément le plus intéressant, mais que des gènes supplémentaires, qui existent dans les k-SIPs, sont aussi à prendre en compte. Les k-SIPs permettent aussi d'identifier des modules métaboliques. Bien que $\overline{w}_d$ et d_G ne permettent pas de les discriminer précisément, nous constatons toutefois que les modules sont en partie constitués par des gènes voisins sur le génomes. Cela indique que la distance de colocalisation est une des distances à utiliser, en association avec d'autres distances codant pour d'autres notions biologiques, afin de découvrir des modules métaboliques de façon automatique.

4.6 Implication de la coexpression de gènes dans les enchaînements de réactions

Dans la section 2.3.2, nous avons vu que l'activité des gènes, lorsqu'elle est coordonnée, indique que ceux-ci participent à une même fonction biologique qui consiste en la transformation d'un substrat en produit. Comment cette coordination transparaît-elle au sein du réseau métabolique d'*E. coli*, en particulier en appliquant SIPPER ?

4.6.1 Génération de $\mathcal{G}_{coexp}$

Nous allons utiliser la même démarche que celle utilisée dans la section 4.5 afin de tester l'hypothèse suivante : les enchaînements de réactions sont-ils liés à des gènes coexprimés ? Nous construisons le modèle intégré d'*E. coli* et nous noterons ce modèle $\mathcal{G}_{coexp}$. Nous y fixons l'ensemble des gènes du génome d'*E. coli* comme l'ensemble d'unités catalytiques. Ainsi l'ensemble des sommets de $\mathcal{G}_{coexp}$ est un ensemble des couples (gène, réaction), où chaque gène est associé aux réactions qu'il catalyse via les enzymes qu'il encode. Afin d'inclure la notion de voisinage de gènes sur le génome à $\mathcal{G}_{coexp}$, nous avons choisi la distance de coexpression, définie à la section 3.3.5. Ainsi chaque arc xy de $\mathcal{G}_{coexp}$ est pondéré par la formule suivante :

$$w_d(xy) = -\ln|cor(g_1, g_2)| \text{ avec } x = (g_1, r),\ y = (g_2, r')$$
$$\text{et } cor(g_1, g_2) \text{ le coefficient de corrélation linéaire entre les séries de } g_1 \text{ et de } g_2$$

Nous avons utilisé les données d'expression de gènes d'une population de bactéries *E. coli* issues de l'expérience référencée *GDS2588* dans la base de données Gene Expression Omnibus [11]. Elle correspond à des mesures d'expression de gènes lors de la croissance anaérobie d'une population de bactéries *E. coli* dans un milieu riche en glucose, c'est-à-dire un milieu propice à la croissance des bactéries. Cette puce a été choisie car elle est l'une des seules puces concerne un maximum de gènes d'*E. coli*. Cela sous-entend que certains gènes n'ont pas été pris en compte par la puce. Cette expérience date de 2006, et certains identifiants de gènes ont changé depuis dans les annotations du réseau métabolique qui date

de 2008. Au final, la puce couvre 87% des gènes d'*E. coli*. Malgré cela, certains gènes n'ont pas de mesures. Les arcs qui ont un poids infini sont retirés de $\mathcal{G}_{coexp}$, impliquant que certains sommets sont isolés des autres, et nous obtenons notre modèle $\mathcal{G}_{coexp}$ qui est composé de 2343 sommets et 11749 arcs qui encodent pour 779 gènes (18,36% du génome). Ces gènes catalysent 1049 réactions via 558 enzymes.

4.6.2 Résultats

Pour chaque k compris entre 1 et 10, nous avons obtenu l'ensemble k-$\mathcal{SIP}_{coexp}$ composé des k-SIPs calculés dans $\mathcal{G}_{coexp}$ entre chaque couple de réactions. Nous avons utilisé pour cela l'algorithme `computeSIP`($\mathcal{G}_{coexp}$, R_1,R_2), R_1 et R_2 étant alors des singletons et avec $R_1 \neq R_2$. Nous avons obtenu 241928 k-SIPs distincts dans chaque k-$\mathcal{SIP}_{coexp}$ soit 46% de k-SIPs en moins que dans $\mathcal{G}_{col}$. En moyenne, un 1-SIP et un 10-SIP contiennent 14,01 ($\pm 7,54$) gènes et 14,61 ($\pm 7,18$) gènes ainsi que 15,75 ($\pm 7,66$) et 19,15 ($\pm 8,35$) réactions. Les k-SIPs obtenus ont en moyenne davantage de gènes et de réactions que dans $\mathcal{G}_{col}$, et la plupart des chemins alternatifs font apparaître en moyenne des nouvelles réactions plutôt que des nouveaux gènes. Nous constatons tout de même que le fait d'ajouter des chemins alternatifs (i.e. de 1 à 10) a toujours un impact limité sur le nombre de gènes et de réactions utilisés dans les k-SIPs.

4.6.2.1 Les k-SIPs de $\mathcal{G}_{coexp}$ constituent des groupes de gènes biologiquement pertinents

Pour un k donné, nous vérifions si les k-SIPs obtenus sont pertinents en calculant, à l'aide de `GO:TermFinder` dans la base Ecocyc, la p-valeur de chaque k-SIP de k-$\mathcal{SIP}_{coexp}$. Pour $k = 1$, 99,99% des 1-SIPs ont une p-valeur significative (i.e. $< 0,05$). Ce résultat tend à montrer que les 1-SIPs obtenus dans $\mathcal{G}_{coexp}$ ne sont pas le fruit du hasard et plus généralement, qu'un k-SIP contient de l'information pertinente. Cela signifie aussi que les gènes (1) dont l'expression est corrélée et (2) qui participent à un enchaînement de réactions sont significatifs ensemble, ce qui semble naturel.

Comme quasiment tous les k-SIPs possèdent une p-valeur, nous vérifions s'il existe une quelconque corrélation entre la mesure $\overline{w}_d$ d'un k-SIP et la p-valeur de l'ensemble des gènes du k-SIP. Le coefficient de corrélation linéaire obtenu est $-0,279$ pour les 1-SIPs, ce qui donne la droite de régression linéaire observable dans la Figure 4.12. Le R^2 de cette droite est de 0,078, ce qui ne la rend pas représentative du nuage de points obtenu. Les conclusions sont les mêmes que lors de l'étude de l'implication du voisinage de gènes dans l'enchaînement de réactions métaboliques : la corrélation entre la valeur $\overline{w}_d$ des 1-SIPs et la p-valeur des ensembles de gènes des 1-SIPs est faible, et le R^2 de la droite de régression linéaire est très petit. La même observation est valable pour les autres k. Il est donc difficile de discriminer des k-SIPs comme étant plus significatifs que d'autres (i.e. avec une plus faible p-valeur) à l'aide de $\overline{w}_d$. Nous allons donc étudier plus particulièrement si les k-SIPs mettent en avant des gènes essentiels, des opérons et des modules de KEGG.

4.6.2.2 Les k-SIPs de $\mathcal{G}_{coexp}$ passent par des gènes essentiels

Les 1-SIPs obtenus dans 1-$\mathcal{SIP}_{coexp}$ mettent en jeu 103 gènes essentiels, soit 77,44% des gènes essentiels métaboliques possibles. Au fur et à mesure que k augmente, le nombre de gènes essentiels mis en jeu dans les k-SIPs augmente en passant à 107 (80,45%) pour 2-$\mathcal{SIP}_{coexp}$, 108 (81,20%) pour 3-$\mathcal{SIP}_{coexp}$, 111 (83,46%) pour 4-$\mathcal{SIP}_{coexp}$ et 112 (84,21%) pour 5-$\mathcal{SIP}_{coexp}$ à 10-$\mathcal{SIP}_{coexp}$.

Bien que leur nombre soit moindre que dans les k-$\mathcal{SIP}_{col}$, nous retrouvons une grande partie des gènes essentiels dans nos k-$\mathcal{SIP}_{coexp}$. Cela semble naturel, puisque les gènes essentiels sont des gènes très étudiés [10, 87, 38, 46], et par conséquent leur expression est mesurée par les puces.

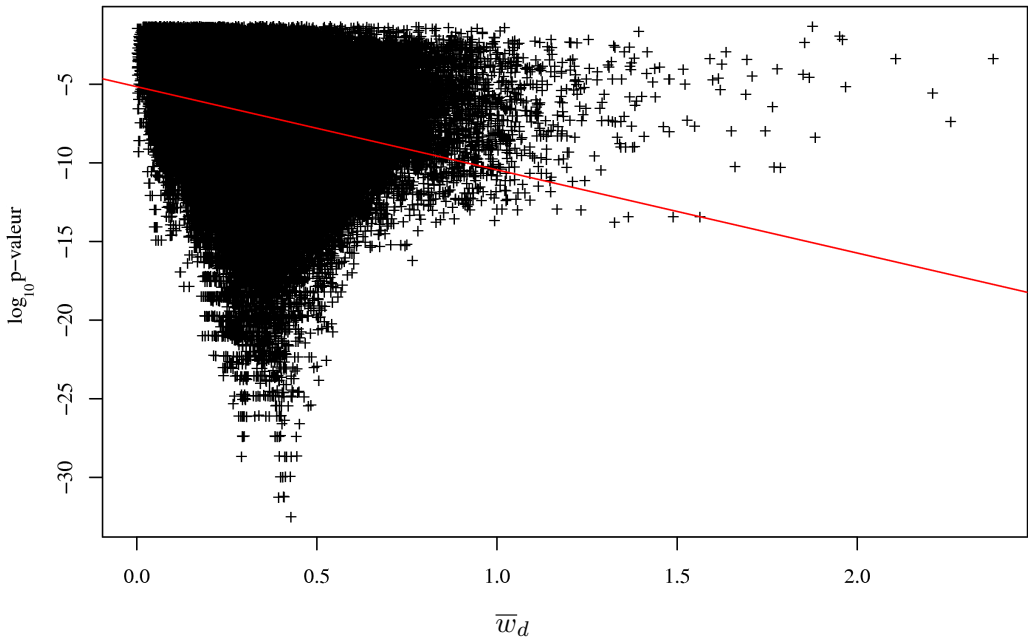


Figure 4.12 – Nuage de points présentant les 1-SIPs de 1-$\mathcal{SIP}_{coexp}$ sous la formes de couples ($\overline{w}_d$,p-valeur). Chaque point représente un 1-SIP. La droite rouge est la droite de régression linéaire du nuage de points. Son R^2 est de 0,078.

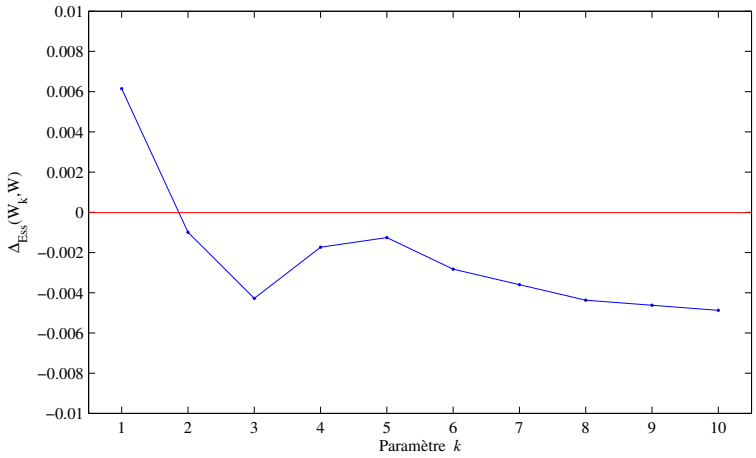


Figure 4.13 – Différence d'utilisation des gènes essentiels intervenant dans k-$\mathcal{SIP}_{coexp}$ et $\mathcal{G}_{coexp}$ en fonction de k. L'étude différentielle est réalisé en utilisant la formule $\Delta_{Ess}(W_k, W)$, avec W_k l'ensemble des gènes intervenant dans les k-SIPs de k-$\mathcal{SIP}_{col}$ et W l'ensemble des gènes intégrés à $\mathcal{G}_{col}$.

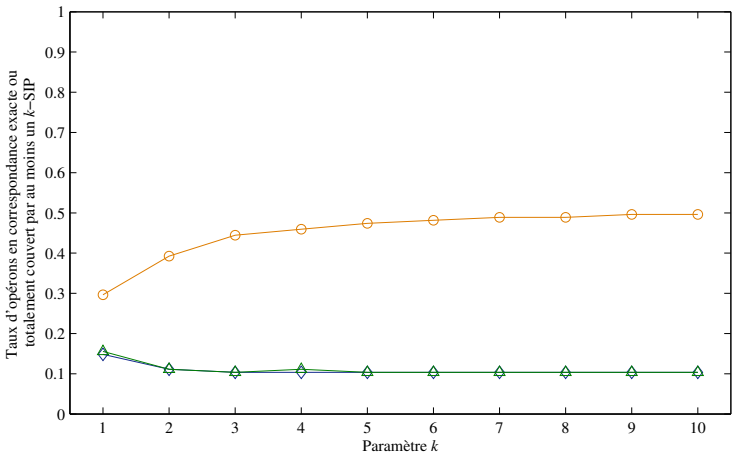


Figure 4.14 – Intérêt opéronique de tous les k-SIPs pour des valeurs de k distinctes dans *E. coli*. La courbe en $\circ$ représente les taux d'opérons qui sont couverts totalement par au moins un k-SIP. La courbe en $\diamond$ (les courbes en $\triangle$ et en $\star$ respectivement) présente le taux d'opérons seuls correspondant exactement à un k-SIP (le taux d'opérons seul ou en couple correspondant exactement à un k-SIP, et le taux d'opérons seuls, en couple ou en triplet correspondant exactement à un k-SIP respectivement).

La distance utilisée dans $\mathcal{G}_{coexp}$ nous permet-elle de distinguer les gènes essentiels des gènes non essentiels par rapport à leur fréquence d'apparition (i.e. leur nombre d'occurrences) dans les k-SIPs ? Nous avons mesuré, comme dans le cas de $\mathcal{G}_{col}$, les valeurs $\Delta_{Ess}(W_k, W)$ (voir Figure 4.13), avec W_k l'ensembles des gènes essentiels intervenant dans k-$\mathcal{SIP}_{coexp}$ et W l'ensemble des gènes dans $\mathcal{G}_{coexp}$. Les valeurs prises par $\Delta_{Ess}(W_k, W)$ sont proches de 0. Dans le cas de $\Delta_{Ess}(W_1, W)$, les gènes essentiels présents dans $1-\mathcal{SIP}_{coexp}$ apparaissent un peu plus fréquemment que les gènes non essentiels. Cependant, cette tendance s'inverse pour les valeurs de k suivantes.

Comme la proportion de gènes essentiels apparaissant dans k-$\mathcal{SIP}_{coexp}$ ne se distingue pas de la proportion de gènes essentiels apparaissant dans $\mathcal{G}_{coexp}$, nous utilisons le test de Wilcoxon pour déterminer si l'hypothèse H_1 suivante est valide : dans k-$\mathcal{SIP}_{coexp}$, la fréquence d'occurrence moyenne des gènes essentiels est distincte de la fréquence d'occurrence moyenne des gènes non-essentiels. Pour cela, nous cherchons à rejeter l'hypothèse nulle H_0 : les distributions des fréquences des gènes essentiels et non essentiels sont similaires. Les résultats de ce test pour k de 1 à 10 sont résumés dans la Table C.1 en annexe C. Dans le cas des 1-SIPs, nous obtenons une p-valeur de 0,0215. Cela signifie que statistiquement que l'hypothèse nulle est rejetable, mais ce test ne nous informe pas sur le fait que la fréquence d'occurrence moyenne des gènes essentiels soit plus ou moins importante que celle des non essentiels. Pour les autres k étudiés, nous obtenons des résultats différents. Les p-valeurs obtenues sont supérieures à 0,05, l'hypothèse nulle est donc fortement probable. Il est donc difficile de discriminer les gènes essentiels à l'aide des k-SIPs en utilisant la distance de coexpression.

4.6.2.3 Certains k-SIPs de $\mathcal{G}_{coexp}$ correspondent à des opérons

Étude relative à k

Comme lors de l'étude de l'impact de la colocalisation de gènes dans le génome sur le réseau métabolique, nous allons étudier ici si les k-SIPs observés correspondent à des opérons et s'il est possible de les discriminer. Pour k entre 1 et 10, l'intérêt opéronique des k-SIPs est résumé dans la Figure 4.14 qui regroupe les taux de correspondance exacte et de couverture totale des opérons par les k-SIPs de $\mathcal{G}_{coexp}$, en fonction de k. Il apparaît que les 1-SIPs couvrent pleinement 29,6% de opérons, tandis que 14,81% (soit 20 opérons) des opérons sont en correspondance exacte avec au moins un 1-SIP (détails disponibles dans la Table C.2, annexe C). Ces taux augmentent et diminuent respectivement à 49,6% et 10,37% (soit 14 opérons) pour $k = 10$. Ces résultats sont plus faibles que ceux observés dans $\mathcal{G}_{col}$. Nous constatons que pour k de 1 à 10, 11 opérons ont au moins une correspondance exacte avec un k-SIP dans tous les ensembles k-$\mathcal{SIP}_{coexp}$. Ce sont les mêmes que nous obtenons que dans l'étude $\mathcal{G}_{col}$. Au total 23 opérons (17,04%) sont en correspondance exacte avec au moins un k-SIP, tous k entre 1 et 10 confondus. Contrairement aux k-SIPs obtenus dans $\mathcal{G}_{col}$ en utilisant la distance de colocation, les k-SIPs obtenus dans $\mathcal{G}_{coexp}$ couvrent exactement et correspondent exactement à moins d'opérons. Ceci est probablement dû au fait que, pour tout k de 1 à 10, l'ensemble k-$\mathcal{SIP}_{coexp}$ est constitué de moins de k-SIPs que k-$\mathcal{SIP}_{col}$.

La Figure 4.14 nous apprend également que certains opérons sont retrouvés en couple par des k-SIPs (courbe $\triangle$). Il s'agit du couple d'opérons composé de *uxuAB* et de *uxaCA* qui correspond exactement à un 1-SIP et du couple composé de *ilvIH* et *ivbL-ilvBN* qui correspond exactement à un 4-SIP (voir Table C.3 de l'annexe C pour détails). Le nombre de couples d'opérons détectés dans cette expérimentation est nettement moins important que celui détecté lors de l'expérimentation réalisée dans $\mathcal{G}_{col}$, toujours lié au fait, entre autres, que pour un k donné, k-$\mathcal{SIP}_{coexp}$ a moins d'éléments que k-$\mathcal{SIP}_{col}$.

Étude relative à $\overline{w}_d$

La mesure $\overline{w}_d$ dépend de la distance d intégrée à $\mathcal{G}_{coexp}$. Afin de mesurer ce changement de distance au niveau de la détection des opérons, nous avons étudié l'intérêt opéronique des k-SIPs en fonction de $\overline{w}_d$ pour des valeurs de k distinctes. La Figure 4.15 résume cette étude pour l'ensemble 1-$\mathcal{SIP}_{coexp}$ qui présente le plus grand taux d'opérons correspondant exactement à un k-SIP chacun (voir Figure 4.14). Elle montre, via la courbe discontinue, que la quasi-totalité des opérons en correspondance exacte avec un 1-SIP le sont pour des valeurs $\overline{w}_d \leq 1$. Contrairement à notre étude relative à $\overline{w}_d$ dans k-$\mathcal{SIP}_{col}$, la valeur $\overline{w}_d = 1$ n'est pas faible dans le cas de k-$\mathcal{SIP}_{coexp}$. En parallèle de cela, le taux de confiance opéronique des 1-SIPs (i.e. les 1-SIPs qui correspondent exactement à un opéron), représenté par la courbe en trait continu, reste très faible. Sur la Figure 4.15 il est confondu avec l'axe des abscisses. Il n'en ressort pas de valeur seuil particulière de $\overline{w}_d$ donnant la garantie d'avoir une proportion raisonnable de 1-SIPs correspondant exactement à un opéron.

Étude relative à la densité génomique

La densité génomique d_G, contrairement à $\overline{w}_d$, ne dépend pas de la distance d intégrée à $\mathcal{G}_{int}$. Elle reste cependant intéressante à étudier sur les k-SIPs de $\mathcal{G}_{coexp}$. La Figure 4.16 résume la pertinence opéronique (voir section 4.4.2) des k-SIPs en présentant, pour chaque classe de densité génomique, la pertinence opéronique des k-SIPs de cette classe. Les 1-SIPs de forte densité génomique (i.e. $\geq 0,9$) sont, pour la moitié d'entre eux, proches d'un opéron avec une pertinence opéronique d'au moins 0,66

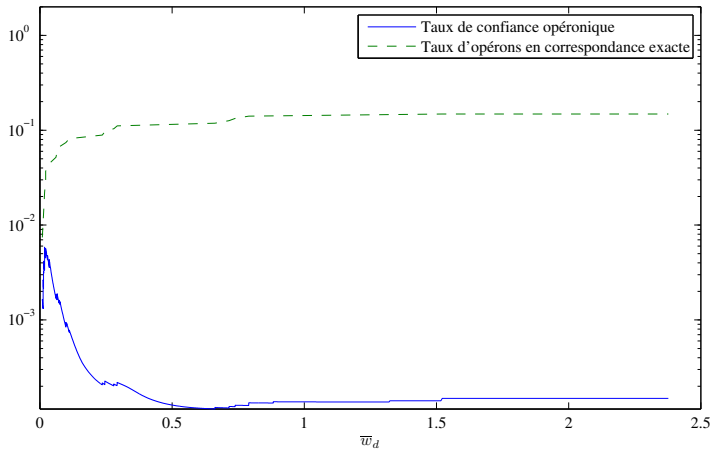


Figure 4.15 – Taux de 1-SIPs qui correspondent exactement à un opéron en fonction de $\overline{w}_d$ dans $\mathcal{G}_{coexp}$. La courbe pleine montre l'évolution de ce taux quand $\overline{w}_d$ est borné par un seuil supérieur marqué sur l'axe des abscisses. En complément, la courbe discontinue représente le taux d'opérons correspondant exactement à un 1-SIP avec un $\overline{w}_d$ borné par cette valeur donnée.

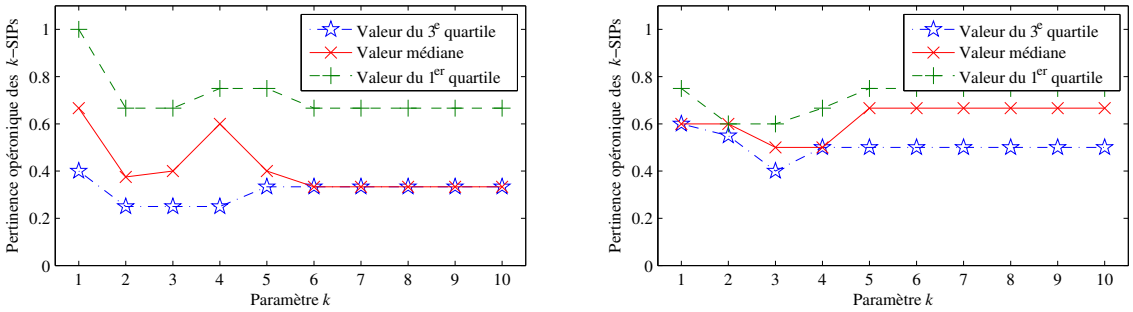
(point $\times$ pour $k = 1$). Quand k augmente, les k-SIPs avec cette densité génomique ne correspond plus vraiment à des opérons. Plus de la moitié des k-SIPs de densité $\geq 0,9$ a une pertinence opéronique inférieure au égale à 0,4 (courbe $\times$), sauf pour $k = 4$ où la pertinence médiane est de 0,6. Trois quarts des k-SIPs de densité génomique comprise entre 0,7 et 0,9 a une pertinence opéronique d'au moins 0,5 (courbe $\star$). Pour n'importe quel k, la pertinence opéronique des k-SIPs est faible pour les densités génomique inférieures à 0,7.

Ces résultats font apparaître une information particulièrement intéressante : les k-SIPs de $\mathcal{G}_{coexp}$ minimisent la distance de coexpression, donc font intervenir des gènes coexprimés, tandis que la mesure de densité, lorsqu'elle est forte, indique que les gènes du k-SIP sont contigus et participent ensemble à un groupe de réactions. Ainsi, la faible pertinence opéronique de ces k-SIPs pour de fortes densités ($\geq 0,9$) quand $k > 1$ indique que les opérons ne sont pas les seuls groupes de gènes contigus dont l'expression est coordonnée, mais qu'il existe d'autres groupes de gènes contigus, n'appartenant pas aux mêmes unités de transcription, qui participent à un enchaînement de réactions métaboliques. Par exemple, nous pourrions avoir affaire à des groupes de gènes dont l'expression dépend de conditions cellulaires non prise en compte dans $\mathcal{G}_{int}$, comme le pH ou la température. Nous pouvons aussi imaginer une régulation complexe de gènes en amont, dont les produits (i.e. protéines) provoquent la transcription des gènes voisins ayant des facteurs de transcription distincts.

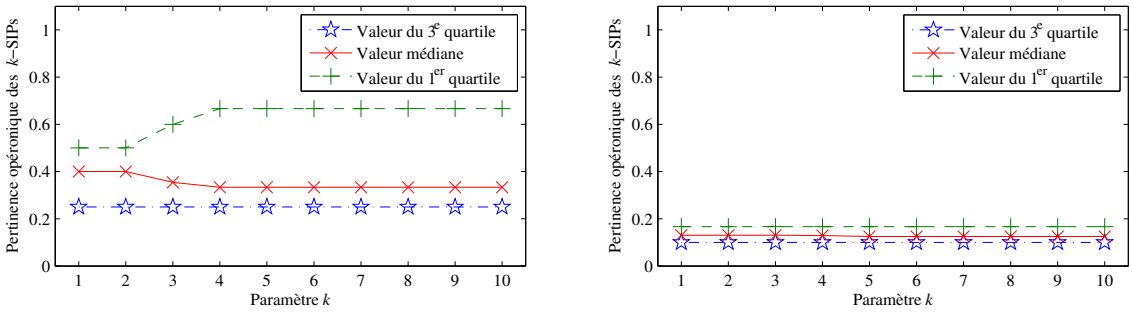
4.6.2.4 Certains k-SIPs de $\mathcal{G}_{coexp}$ correspondent à des modules métaboliques

Étude relative à k

La Figure 4.17 résume l'intérêt modulaire des k-SIPs en regroupant la couverture totale et les taux de correspondances exactes entre modules et k-SIPs. Nous avons trouvé que 21 modules (21.21%) correspondent exactement à des 1-SIPs (voir courbe $\diamond$ dans la Figure 4.17 et la Table C.4 dans l'annexe C pour une liste précise), et que 41,41% des modules sont couverts totalement par au moins un 1-SIP (courbe



(a) $0.9 \leq d_G \leq 1.0$ (b) $0.7 \leq d_G < 0.9$



(c) $0.3 \leq d_G < 0.7$ (d) $0.0 \leq d_G < 0.3$

Figure 4.16 – Évolution de la pertinence opéronique des k-SIPs groupés par classe de densité génomique pour $1 \leq k \leq 10$. L'évolution de chaque classe de densité génomique : (a) $0.9 \leq d_G \leq 1.0$, (b) $0.7 \leq d_G < 0.9$, (c) $0.3 \leq d_G < 0.7$ et (d) $0.0 \leq d_G < 0.3$, est résumée par trois courbes dans chaque cas : les courbes $\times$, $+$ et $\star$ représentent respectivement l'évolution de la valeur médiane, la valeur du 1^{er} et du 3^e quartile de la pertinence des k-SIPs de densité d.

$\circ$ dans la Figure 4.7). Ces deux taux tombent et montent respectivement à 15,15% et 45,45% quand $k = 10$. De plus, 14.14% des modules de KEGG ont toujours une correspondance exacte pour tous les k de 1 à 10. Nous observons aussi 9 couples (courbe $\triangle$ dans la Figure 4.17) et 1 triplet (courbe $\star$ dans la Figure 4.17) de modules qui correspondent exactement à des k-SIPs pour $k \geq 1$ (voir la Table C.5 en annexe C pour plus de détails). Bien que le nombre de modules en correspondance exacte avec des k-SIPs soit moins important dans $\mathcal{G}_{coexp}$ que dans $\mathcal{G}_{col}$, le nombre de modules qui sont retrouvés associés en couples reste similaire. Dans la Figure 4.17, les courbes $\triangle$ et $\star$ montrent l'augmentation du taux de modules identifiés quand les couples et triplets de modules sont considérés en plus des modules seuls.

Étude relative à $\overline{w}_d$

Pour k de 1 à 10, afin de déterminer si certains k-SIPs de k-$\mathcal{SIP}_{coexp}$ discriminent des modules de KEGG, nous étudions, dans la Figure 4.18, les taux de confiance modulaire et de correspondance exacte des 1-SIPs avec les modules en fonction d'une borne maximum de $\overline{w}_d$ (axe des abscisses). Il n'apparaît pas de valeur de $\overline{w}_d$ qui mette en évidence des modules. Ceux-ci apparaissent pour des valeurs de $\overline{w}_d$ comprises entre 0 et $1,522$ comme le montre la courbe discontinue. La Table C.4 de l'annexe C

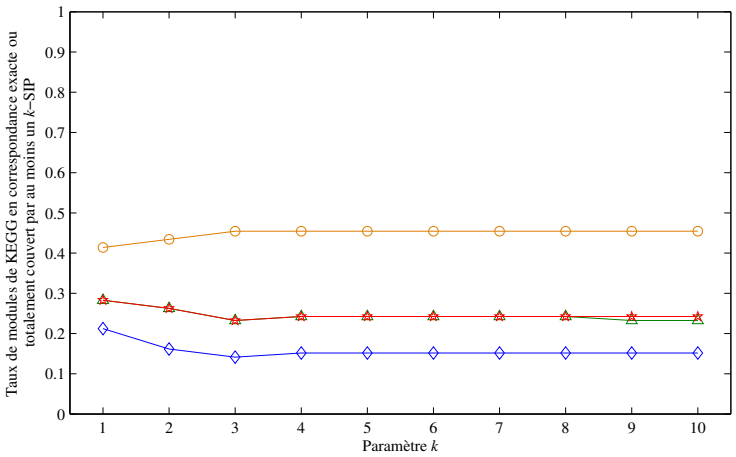


Figure 4.17 – Intérêt modulaire de tous les k-SIPs pour des k distincts dans *E. coli*. La courbe en $\circ$ représente les taux de modules de KEGG couverts totalement par au moins un k-SIP. La courbe en $\diamond$ (les courbes en $\triangle$ et $\star$ respectivement) résume le taux de modules seuls (en couple ou en triplet respectivement) qui correspondent exactement à des k-SIPs.

détaille ces différentes valeurs pour chaque modules en correspondance exacte avec un 1-SIP. Ainsi, w_d ne permet pas de mettre en évidence les k-SIPs correspondant aux modules métaboliques dans $\mathcal{G}_{coexp}$.

Étude relative à la densité génomique

Certains k-SIPs correspondent exactement à des modules de KEGG. Nous avons étudié la façon dont le critère $\overline{w}_d$ les met en valeur, mais comment le critère de densité génomique d_G, qui ne dépend pas de la distance de coexpression utilisée dans $\mathcal{G}_{coexp}$, les caractérise-t-il ? Étant donné un k fixé, nous utilisons la pertinence modulaire des k-SIPs, décrite dans la section 4.4.2, afin de les étudier en fonction de leur densité (voir la Figure 4.19).

Lorsque nous étudions les 1-SIPs de densité génomique $\geq 0,9$, nous observons que leur pertinence modulaire est faible, c'est-à-dire qu'elle est inférieure à 0,4 pour les trois quarts d'entre eux, quel que soit k compris entre 1 et 10 (voir courbe $+$ dans la Figure 4.19(a)). Dans le cas des 1-SIPs de densité comprise entre 0,7 et 0,9, plus de la moitié d'entre eux ont une pertinence modulaire supérieure à 0,6. Cependant, la pertinence des k-SIPs diminue très vite pour atteindre 0 pour $k \geq 3$. Cela indique que les 1-SIPs de $\mathcal{G}_{coexp}$ de densité génomique comprise entre 0,7 et 0,9 ressemblent à des modules métaboliques de KEGG, mais que rajouter des chemins alternatifs change leur densité génomique, indiquant que les gènes dont l'expression est corrélée ne sont pas forcément des gènes voisins. Lorsque nous nous intéressons aux k-SIPs de densité génomique inférieure à 0,7 (Figure 4.19(c) et Figure 4.19(d)), les k-SIPs, quel que soit k, n'ont pas une très grande pertinence modulaire.

4.6.3 Résumé

Tout d'abord, quasiment tous les k-SIPs de $\mathcal{G}_{coexp}$ que nos obtenons sont cohérents avec l'information contenue dans la base de connaissance Ecocyc. Ensuite, la distance de coexpression n'est pas plus adaptée que la distance de colocalisation pour mettre en valeur les gènes essentiels. Cela va dans le

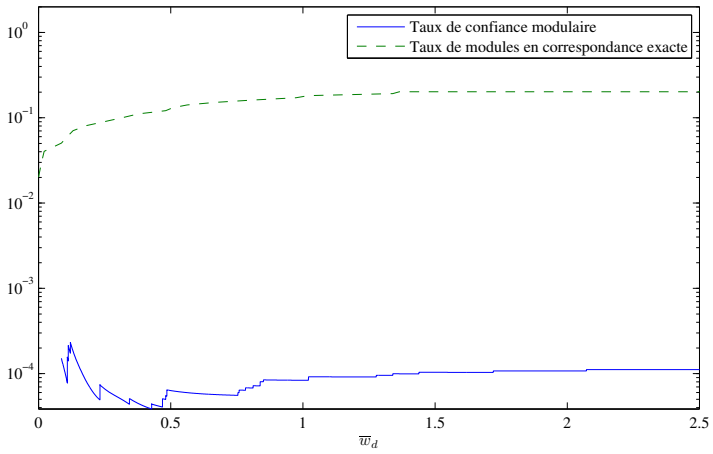


Figure 4.18 – Taux de 1-SIPs qui correspondent exactement à un module de KEGG en fonction de $\overline{w}_d$ dans *E. coli*. La courbe pleine montre l'évolution de ce taux quand $\overline{w}_d$ est borné de façon supérieure par une valeur donnée sur l'axe des abscisses. La courbe discontinue représente le taux de modules en correspondance exacte avec un k-SIP pour la borne supérieure de $\overline{w}_d$ donnée.

sens de del Rio *et al.* [32] qui indiquent que les mesures de centralité, qui indiquent l'importance [3] d'un sommet dans un graphe, ne sont pas adaptées à la détermination des gènes essentiels.

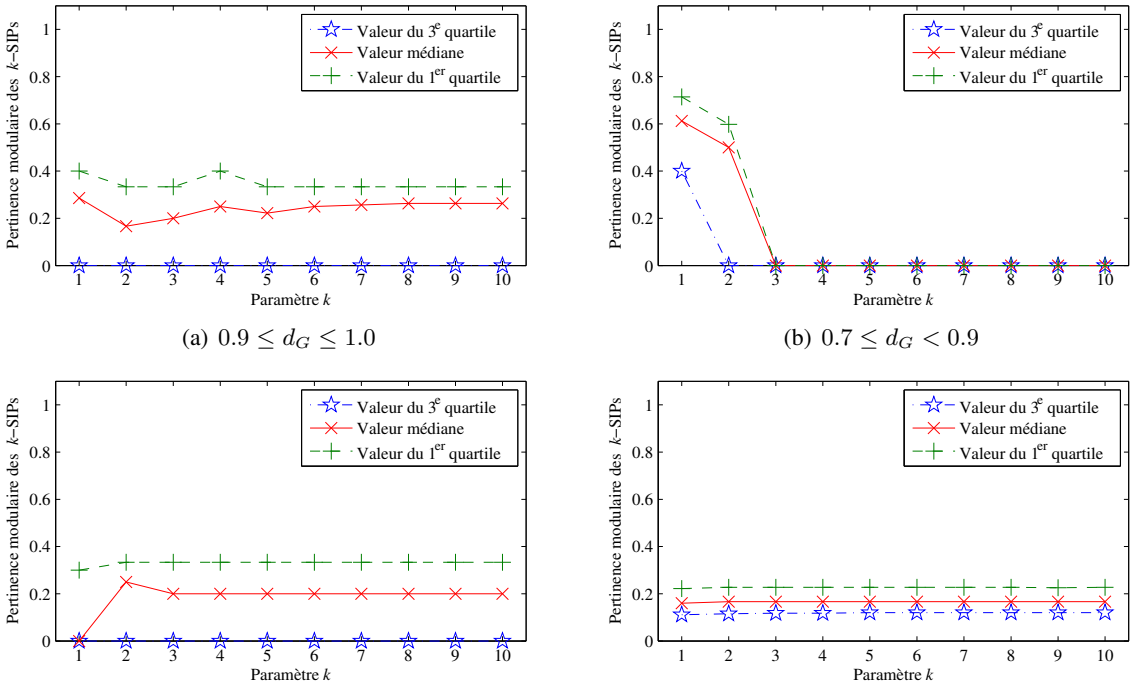
Il est admis dans les systèmes bactériens que l'expression de gènes contigus s'effectue de façon coordonnée, c'est-à-dire que les niveaux d'expression de ces gènes sont liés. En utilisant cette remarque comme base de notre hypothèse biologique, nous espérions retrouver autant d'opérons en utilisant la distance de coexpression qu'en utilisant la distance de colocalisation ; mais ce n'est pas le cas. Toutefois, nous observons, d'une part, que nous avons moitié moins de k-SIPs dans $\mathcal{G}_{coexp}$ que dans $\mathcal{G}_{col}$, et d'autre part, que nous mettons en évidence que les k-SIPs de forte densité génomique ($\geq 0,9$) font parfois apparaître des groupes de gènes contigus (à cause de la forte densité des k-SIPs) qui ne correspondent pas à des opérons. L'expression des gènes de ces groupes est probablement liée à des conditions cellulaires non prises en compte dans $\mathcal{G}_{int}$, comme le pH ou la température ou bien une régulation complexe de gènes qui provoque la transcription de ces gènes voisins ayant des facteurs de transcription distincts.

D'un point de vue modulaire, la distance de coexpression n'est pas adaptée non plus pour discriminer les modules métaboliques. Cependant, l'usage de cette distance avec le critère de densité génomique permet d'en discriminer certains, quand la densité du k-SIP est entre 0,7 et 0,9 pour $k = 1$. Cette observation renforce l'idée que les modules métaboliques peuvent être identifiés en utilisant non pas une seule distance pour pondérer $\mathcal{G}_{int}$, mais plusieurs distances.

4.7 Conclusion

Dans cette partie, nous avons testé deux hypothèses biologiques, sur *E. coli*, qui nous ont conduits, chacune, à un modèle intégré $\mathcal{G}_{int}$, nommé $\mathcal{G}_{col}$ dans la première étude (voir section 4.5) et $\mathcal{G}_{coexp}$ dans la

3. Par exemple, la *betweeness centrality* indique à quel point un sommet intervient dans des chemins, ou le *degré de connectivité* indique à quel point un sommet est connecté aux autres.



(a) $0.9 \leq d_G \leq 1.0$ (b) $0.7 \leq d_G < 0.9$

(c) $0.3 \leq d_G < 0.7$ (d) $0.0 \leq d_G < 0.3$

Figure 4.19 – Évolution de la pertinence modulaire des k-SIPs de $\mathcal{G}_{coexp}$ groupés par classe de densité génomique pour $1 \leq k \leq 10$. L'évolution de chaque classe de densité génomique : (a) $0.9 \leq d_G \leq 1.0$, (b) $0.7 \leq d_G < 0.9$, (c) $0.3 \leq d_G < 0.7$ et (d) $0.0 \leq d_G < 0.3$, est résumée par trois courbes dans chaque cas : les courbes $\times$, $+$ et $\star$ représentent respectivement l'évolution de la valeur médiane, la valeur du 1^{er} et du 3^e quartile de la pertinence modulaire des k-SIPs de densité d.

seconde (voir section 4.6). Nous les avons ensuite étudiées à l'aide `SIPPER` et nous avons obtenu, pour $\mathcal{G}_{col}$ et $\mathcal{G}_{coexp}$, un ensemble de k-SIPs dont nous avons évalué la pertinence biologique.

Tout d'abord, nous observons que tous les k-SIPs de $\mathcal{G}_{col}$ et $\mathcal{G}_{coexp}$ sont pertinents avec la connaissance disponible dans la base de connaissance Ecocyc. D'ailleurs, dans $\mathcal{G}_{col}$, la moitié d'entre eux correspondent à des opérons ou à des modules métaboliques. Un k-SIP de $\mathcal{G}_{col}$ représente, dans le réseau métabolique, une chaîne de réactions catalysées par des enzymes, produits par des gènes proches sur le génome. Un k-SIP de $\mathcal{G}_{coexp}$ représente, dans le réseau métabolique, une chaîne de réactions catalysées par des enzymes dont les gènes codants s'expriment de la façon la plus corrélée possible.

Nous avons ensuite essayé de mettre en évidence les gènes essentiels. Nous avons constaté qu'une grande partie d'entre-eux intervient dans au moins un k-SIP (pour k de 1 à 10, 94% à 98% dans $\mathcal{G}_{col}$ et 80% à 84% dans $\mathcal{G}_{coexp}$). Il convient de remarquer que, contrairement à $\mathcal{G}_{col}$, seule une partie des gènes métaboliques est vraiment prise en compte dans $\mathcal{G}_{coexp}$. En effet, bien que nous ayons utilisé une puce couvrant le plus de gènes possibles, la moitié des gènes de *E. coli* n'a pas de mesure d'expression. Dans $\mathcal{G}_{coexp}$ et par construction, les sommets associés à ces gènes n'ont donc pas de voisins, ce qui rend impossible la recherche de k-SIPs utilisant ces gènes. Cela explique que le nombre de k-SIPs est moitié moindre dans $\mathcal{G}_{coexp}$ que dans $\mathcal{G}_{col}$. Malgré cela, les gènes essentiels sont toujours bien représentés. Dans $\mathcal{G}_{col}$ et $\mathcal{G}_{coexp}$, nous n'avons pas pu discriminer les gènes essentiels des gènes non-essentiels à l'aide

des k-SIPs. Ce résultat conforte celui de del Rio *et al.* [32], qui indiquent que les mesures de centralité, comme le nombre d'occurences d'un gène dans un ensemble de chemins, ne sont pas les plus adaptées pour déterminer les gènes essentiels d'un organisme.

Nous avons également cherché à discriminer, parmi les k-SIPs, certaines entités biologiques qui sont les opérons et les modules métaboliques de KEGG. Nous avons réalisé cela à l'aide des mesures dédiées de coefficient de voisinage $\overline{w}_d$ et de densité génomique d_G.

Nous avons remarqué, d'après les résultats obtenus sur $\mathcal{G}_{col}$, que l'utilisation de la distance de colocalisation se prête bien à la recherche d'opérons métaboliques. Les mesures de coefficient de voisinage $\overline{w}_d$ et plus particulièrement de densité génomique d_G nous permettent d'identifier une partie d'entre eux, et le paramètre k, qui définit le nombre de chemins alternatifs d'un k-SIP, nous permet d'obtenir des variations autour de ces opérons. En effet, non seulement nous trouvons des opérons de façon exacte, mais aussi des opérons associés à d'autres gènes proches. Cela suggère que ces gènes additionnels sont liés à l'opéron par une dépendance fonctionnelle. Dans l'étude de $\mathcal{G}_{coexp}$, nous n'obtenons pas de résultats aussi clairs. La discrimination de k-SIPs opéroniques à l'aide de $\overline{w}_d$ n'est pas convaincante par rapport aux résultats obtenus dans $\mathcal{G}_{col}$. De même, la densité génomique, quand elle est supérieure à 0,7, discrimine plus difficilement les k-SIPs correspondant à des opérons dans $\mathcal{G}_{coexp}$ que dans le cas de $\mathcal{G}_{col}$, mais fait apparaître une information importante : il existe des groupes de gènes contigus qui sont coexprimés et qui ne sont pas des opérons. Cela signifie que des gènes voisins peuvent être coordonnés au niveau de leur expression sans appartenir à la même unité de transcription. Cette remarque renforce l'observation constatée dans $\mathcal{G}_{col}$ où nous avons des enchaînements de réactions faisant intervenir un opéron et quelques gènes voisins additionnels. Ces groupes de gènes particuliers devront être étudié précisément par les biologistes.

Dans le cas des modules métaboliques, l'utilisation de la distance de colocalisation seule (i.e. dans $\mathcal{G}_{col}$) se prête moins bien à l'identification des modules. Étant donné qu'ils sont définis manuellement, à partir de plusieurs de plusieurs sources d'informations (études phylogéniques, réactions catalysées par des gènes coexprimés ou appartenant à la même unité de transcription, ...), il est difficile de trouver une distance permettant de discriminer ces k-SIPs dits modulaires. Nous retrouvons cependant certains k-SIPs qui correspondent exactement à des modules avec un $\overline{w}_d$ petit, et nous avons remarqué également que certains k-SIPs modulaires utilisent quelques gènes voisins. Afin de les détecter, peut-être ne devrions nous pas utiliser une distance reposant sur une seule information biologique, mais, comme le suggère la définition même d'un module, une distance reposant sur un ensemble de notions et informations biologiques (phylogénie, trancription, expression des gènes, ...) Les résultats observés sur $\mathcal{G}_{coexp}$ vont d'ailleurs dans ce sens. L'utilisation de la distance de coexpression, suivie de l'analyse en fonction de la densité génomique fait apparaître que les 1-SIPs de $\mathcal{G}_{coexp}$ avec une densité génomique entre 0,7 et 0,9 correspondent bien à des modules métaboliques. Ainsi le cumul de l'information de coexpression, induite par la pondération de $\mathcal{G}_{coexp}$, et l'information de proximité génomique, induite par une densité génomique assez forte, fait apparaître la nécessité d'utiliser plusieurs distances codant chacune pour une information biologique afin d'identifier des modules métaboliques.

Les résultats que nous avons obtenus, en particulier ceux à propos des k-SIPs de forte densité génomique, nous permettent d'identifier des opérons métaboliques, mais aussi des groupes de gènes opéroniques associés à d'autres gènes dont l'expression est corrélée, pour ne pas dire coordonnée. Fort de ces résultats, nous proposons d'évaluer l'apport de l'intégration de données métaboliques et d'expression de gènes à la génomique comparative. Nous allons donc chercher des intervalles communs de gènes entre deux génomes en fonction de l'information métabolique ou d'expression ajoutée.

CHAPITRE 5

Contribution de notre méthode intégrative à la génomique comparative

5.1 Introduction

Ce chapitre détaille les travaux effectués dans [5] et présente l'intérêt de l'intégration de données *omiques* telle que proposée précédemment dans les méthodes développées pour la comparaison de génomes et le transfert d'annotations de groupes de gènes conservés. Dans ce chapitre, nous présenterons d'abord ce qu'est la comparaison de génomes au niveau de leur séquence de gènes, en particulier la notion d'intervalles communs à deux génomes. Puis nous développerons une méthode de comparaison de génomes basée sur les intervalles communs qui intègrent des informations *omiques*. Nous étudierons enfin l'application de cette méthode en comparant des opérons aux intervalles de gènes communs aux génomes d'*E. coli* et *V. cholerae*.

5.1.1 La comparaison de deux génomes

Dans ce chapitre, les génomes sont décrits et comparés comme des séquences de gènes. La comparaison de deux génomes consiste alors à étudier les gènes communs ou non qui existent dans ces deux génomes. Afin de mesurer les différences et similitudes, des opérations élémentaires, dites de réarrangement génomique, qui permettent de transformer un génome G_0 en un autre G_1, ont été définies. Ces opérations permettent également d'établir une distance séparant deux génomes, qui est généralement le nombre minimum d'opérations transformant l'un en l'autre. L'idée derrière cette approche est que deux espèces proches par la distance entre leurs génomes (i.e. séparées par un nombre d'opérations de réarrangement génomique faible) sont aussi proches par d'autres aspects biologiques comme leur faculté d'adaptation ou un comportement biologique particulier.

Afin de définir précisément ces opérations, nous décrivons un génome comme suit.

Définition 26 (Représentation d'un génome)**.** *Un génome est un ensemble de chromosomes. Les bactéries ont généralement un génome composé d'un seul chromosome comme c'est le cas pour E. coli. Ce n'est cependant pas toujours le cas : le génome de V. cholerae est lui composé de deux chromosomes.*

Définition 27 (Représentation d'un chromosome)**.** *Chaque chromosome est représenté par une séquence de gènes. Ainsi, un chromosome chr s'écrira de la sorte : $chr = g_0 g_1 g_2 \dots g_n$ avec g_i, $0 \leq i \leq n$ le i-ème gène du chromosome. Quand le chromosome est circulaire, sa séquence est circulaire. Les gènes*

peuvent être signés (avec $+$ et $-$), afin d'indiquer sur quel brin du chromosome ils se trouvent. Dans notre application, les chromosomes seront toutefois formés de gènes non-signés.

Les principales opérations de réarrangement génomique

Les principales opérations de réarrangement génomique au sein d'un chromosome sont les suivantes :
- la suppression, qui consiste à retirer une suite de gènes du chromosome, par exemple $chr_1 = g_0\ \mathbf{g_1}\ \mathbf{g_2}\ g_3$ devient $chr_1 = g_0\ g_3$,
- l'insertion, qui consiste à ajouter une suite de gènes dans le chromosome, par exemple $chr_1 = g_0\ g_1\ g_2$ devient $chr_1 = g_0\ g_1\ \mathbf{g_a}\ \mathbf{g_b}\ g_2$,
- l'inversion, qui consiste à inverser le sens de lecture d'une suite de gènes du chromosome, par exemple $chr_1 = g_0\ \mathbf{g_1}\ \mathbf{g_2}\ g_3$ devient $chr_1 = g_0\ \mathbf{g_2}\ \mathbf{g_1}\ g_3$,
- la transposition, qui consiste à déplacer une suite de gènes sur le chromosome, par exemple $chr_1 = g_0\ \mathbf{g_1}\ \mathbf{g_2}\ g_3$ devient $chr_1 = g_0\ g_3\ \mathbf{g_1}\ \mathbf{g_2}$,
- la duplication, qui consiste à recopier une suite de gènes d'un chromosome à un autre endroit dans le chromosome, par exemple $chr_1 = g_0\ g_1\ g_2\ g_3$ devient $chr_1 = g_0\ g_1\ g_2\ g_3\ \mathbf{g_1}\ \mathbf{g_2}$.

Les opérations décrites ci-dessus ont lieu au sein d'un chromosome, mais il en existe aussi entre les chromosomes d'un génome :
- la translocation, qui consiste à échanger une suite de gènes d'un premier chromosome avec une suite de gènes, potentiellement vide, d'un second chromosome, par exemple $chr_1 = g_0\ \mathbf{g_1}\ \mathbf{g_2}\ g_3$ et $chr_2 = g_a\ \mathbf{g_b}\ \mathbf{g_c}$ deviennent $chr_1 = g_0\ \mathbf{g_b}\ \mathbf{g_c}\ g_3$ et $chr_2 = g_a\ \mathbf{g_1}\ \mathbf{g_2}$,
- la fusion, qui consiste à regrouper deux chromosomes en un seul, par exemple $chr_1 = g_0\ g_1\ g_2$ et $chr_2 = \mathbf{g_a}\ \mathbf{g_b}$ deviennent $chr = g_0\ g_1\ g_2\ \mathbf{g_a}\ \mathbf{g_b}$
- la fission, qui consiste à diviser un chromosome en deux chromosomes distincts, par exemple $chr = g_0\ g_1\ g_2\ \mathbf{g_a}\ \mathbf{g_b}$ devient $chr_1 = g_0\ g_1\ g_2$ et $chr_2 = \mathbf{g_a}\ \mathbf{g_b}$.

Toute ces opérations permettent de construire un nouveau génome à partir d'un génome donné. La propriété d'adjacence de gènes au sein d'une suite de gènes est la propriété clé de ces réarrangements génomiques. Cependant, nous n'allons pas utiliser ces opérations élémentaires pour mesurer la distance qui sépare deux génomes, mais nous allons tenter de comprendre pourquoi, en comparant deux génomes bactériens, certains groupes de gènes adjacents n'ont pas été modifiés. Bergeron et Stoye [15] décrivent plusieurs notions, dont les intervalles communs [102] et les intervalles conservés [15], comme moyens de décrire et mesurer la conservation de l'adjacence des gènes au sein d'un génome. Dans la suite de ce chapitre, nous nous intéressons en particulier aux intervalles communs, car ils sont plus généraux que les intervalles conservés.

Par convention, afin de simplifier la présentation des différentes notions, nous allons travailler sur des génomes unichromosaux. Lorsque nous ferons référence à un génome, nous ferons référence à son unique chromosome.

Intervalles communs

La notion d'intervalles communs repose sur la notion de permutation.

Définition 28 (Famille de gènes et alphabet)**.** *Une famille de gènes regroupe des gènes déclarés comme homologues. Elle est représentée par une étiquette, qui est un entier positif. La notation $|g|$ indique la famille à laquelle le gène g, identifié par sa place sur le chromosome, appartient. Lorsque la famille d'un gène g contient plus de deux éléments, nous dirons que g est dupliqué. Un ensemble de familles de gènes*

est représenté par un ensemble $\mathcal{F}$ d'entiers positifs appelé alphabet*. Dans un génome $\mathcal{G}$ l'ensemble des familles de gènes présentes sera noté $\mathcal{F}_\mathcal{G}$.*

Définition 29 (Chaîne et sous-chaîne)**.** *Soit un alphabet $\mathcal{F}$. Une chaîne est une suite d'éléments de $\mathcal{F}$. Étant donnée une chaîne s, nous appelons sous-chaîne de s une chaîne présente dans s. Par exemple, soient l'alphabet $\mathcal{F} = \{1, 2, 3, 4\}$ et la chaîne $s = 1\ 2\ 1\ 3\ 4\ 1$. La chaîne $s' = 3\ 4\ 1$ est une sous-chaîne de s. Son alphabet, noté $\mathcal{F}_{s'}$, est égal à $\{1, 3, 4\}$ et est inclus dans $\mathcal{F}$.*

Définition 30 (Longueur d'une chaîne)**.** *Soient un alphabet $\mathcal{F}$ et s une chaîne définie sur $\mathcal{F}$. La longueur de s, notée $||s||$, est le nombre d'éléments de la chaîne.*

Définition 31 (Permutation)**.** *Soit un alphabet $\mathcal{F}$. Une* permutation *est une chaîne définie sur $\mathcal{F}$ pour laquelle chaque élément de $\mathcal{F}$ apparaît une seule et unique fois.*

Il est maintenant possible de définir précisément les intervalles communs [102].

Définition 32 (Intervalle commun)**.** *Étant données deux permutations P_0 et P_1 définies sur $\mathcal{F}$, un* intervalle commun *à P_0 et P_1 est une sous-chaîne s de P_0 telle que P_1 contient une permutation de $\mathcal{F}_s$.*

Les intervalles conservés [15], que nous ne traiterons pas dans cette thèse, sont une version des intervalles communs définis sur un génome signé.

Couplage de gènes et $\mathcal{M}$-élagage

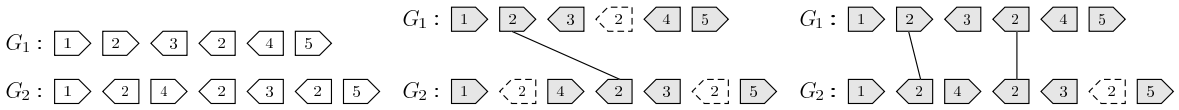
Traitement des gènes dupliqués La notion de permutation, qui sert de base aux intervalles communs et conservés, repose sur l'idée qu'il n'existe qu'une seule occurrence (i.e. un gène) de chaque famille de gènes dans chaque génome. Or ce n'est pas le cas. Les gènes dupliqués peuvent être pris en compte en choisissant une manière de coupler les gènes de deux génomes :

Définition 33 (Couplage de gènes)**.** *Soient deux génomes $G_0 = g_0^0 g_1^0 \dots g_{n_0}^0$ et $G_1 = g_0^1 g_1^1 \dots g_{n_1}^1$. Un couplage (de gènes) $\mathcal{M}$ de (G_0, G_1) est un ensemble de paires (g_i^0, g_j^1) tel que $|g_i^0| = |g_j^1|$ et que, pour toutes paires distinctes $(g_i^0, g_j^1) \in \mathcal{M}$ et $(g_{i'}^0, g_{j'}^1) \in \mathcal{M}$, $g_i^0 \neq g_{i'}^0$ et $g_j^1 \neq g_{j'}^1$.*

Traitement des gènes n'apparaissant que dans un génome En renommant les gènes de chaque couple de $\mathcal{M}$ avec une étiquette unique au couple, nous nous rapprochons de la notion de permutation (l'ensemble des nouvelles étiquettes constitue alors un alphabet). Cependant, le couplage seul ne suffit pas. En effet, même avec un réétiquetage des gènes couplés, les génomes G_0 et G_1 peuvent chacun contenir des gènes qui n'existent pas dans l'autre génome. Afin de transformer le couple de génomes (G_0, G_1) en un couple de permutations (P_0, P_1), il faut retirer de G_0 et de G_1 tous les gènes non couplés. C'est ce que nous appelons le *$\mathcal{M}$-élagage*. En utilisant le couplage de gènes $\mathcal{M}$ et le *$\mathcal{M}$-élagage*, il est possible de définir des modèles de couplage qui permettent ainsi de transformer deux génomes quelconques possédant des gènes dupliqués en deux permutations.

Modèle de couplage de gènes Des modèles classiques de couplages de gènes ont déjà été décrits dans la littérature. En voici une description qui est aussi illustrée par la Figure 5.1 :

- Le *modèle de couplage exemplaire* [93]. Pour chaque famille de gènes f présente à la fois dans G_0 et G_1, le couplage $\mathcal{M}$ ne peut contenir qu'un gène de la famille f. Nous réalisons ensuite le $\mathcal{M}$-élagage de (G_0, G_1) pour obtenir (G_0^E, G_1^E), deux permutations. Nous appelons le triplet $(G_0^E, G_1^E, \mathcal{M})$ un *couplage exemplaire* de (G_0, G_1).



(a) Deux génomes avec des gènes dupliqués (b) Un couplage *exemplaire* possible (c) Un couplage *maximum* possible

Figure 5.1 – Exemple de couplage de gènes sous les modèles de *couplage exemplaire* et *maximum*. (a) Soit deux génomes G_1 et G_2 tels que seul un gène de la famille étiquetée 2 est dupliqué dans les génomes. (b) Le modèle de *couplage exemplaire* n'autorise à conserver qu'une seule occurrence du gène 2. (c) Dans le modèle du *couplage maximum*, deux occurrences du gène 2 sont conservées. Les gènes tracés avec des lignes discontinues sont des gènes qui ont été $\mathcal{M}$*-élagués* lors du couplage.

– Le *modèle de couplage maximal* [101]. Dans ce modèle, nous maximisons le nombre de couples présents dans le couplage $\mathcal{M}$. Nous réalisons ensuite le $\mathcal{M}$-élagage de (G_0, G_1) pour obtenir (G_0^M, G_1^M). Nous appelons le triplet $(G_0^M, G_1^M, \mathcal{M})$ un *couplage maximal* de (G_0, G_1).

5.2 Méthode

5.2.1 Le problème du nombre maximum d'intervalles communs

Nous voulons expliquer, à l'aide d'informations *omiques* supplémentaires, pourquoi des groupes de gènes sont conservés entre G_0 et G_1, deux génomes pouvant être circulaires ou linéaires et possédant des gènes dupliqués. La section précédente a présenté la notion d'intervalle commun comme une entité appropriée pour définir ces groupes de gènes conservés[6].

Afin de comparer au mieux deux génomes G_0 et G_1, Angibaud *et al.* [6] ont choisi de calculer le nombre d'intervalles communs à deux génomes de contenus différents et avec gènes dupliqués. Cependant, étant donnés un modèle de couplage $\mathcal{M}$ et deux génomes avec des gènes dupliqués, coupler les gènes selon $\mathcal{M}$ de telle sorte que le nombre d'intervalles communs obtenus par ce couplage soit maximum est un problème difficile. Le problème de décision correspondant est NP-complet [25]. Angibaud *et al.* ont présenté différentes approches pour calculer ce nombre de manière exacte [6], mais aussi de façon non-exacte [7] . Angibaud *et al.* ont également développé le logiciel `Match&Watch` [6] qui calcule ce nombre maximum d'intervalles communs à deux génomes de contenus différents et avec gènes dupliqués et qui énumère également ces intervalles. Ce logiciel fonctionne comme décrit dans l'Algorithme 8.

Algorithme 8 `Match&Watch`

ENTRÉES: Deux génomes G_0 et G_1
Un modèle de couplage $\mathcal{M}$ (couplage *exemplaire* ou *maximum*)
SORTIES: Un ensemble des intervalles communs entre G_0 et G_1

1: Soit $\mathcal{H}$ l'ensemble des homologies de gènes existant dans G_0 et G_1 calculées par le logiciel `Inparanoid`
2: Renommer les gènes de G_0 et G_1 en accord avec leur homologie dans $\mathcal{H}$
3: Calculer un couplage de gènes sous le modèle de $\mathcal{M}$ qui vise à maximiser le nombre d'intervalles communs
4: Renommer et $\mathcal{M}$-élaguer les gènes en accord avec le couplage choisi.
5: Générer la liste des intervalles communs.

La première étape de `Match&Watch` (ligne 1) consiste à calculer les gènes homologues, c'est-à-dire les gènes qui sont considérés comme similaires. Le logiciel `Inparanoid` [84] a été utilisé pour cette tâche. Ce logiciel permet de créer des familles de gènes homologues en réalisant une analyse de séquence des protéines issues des gènes à l'aide de `BlastP` [3]. Ensuite, les gènes sont annotés avec une étiquette similaire s'ils appartiennent à la même famille (ligne 2).

Quand un génome contient des gènes dupliqués, les gènes sont couplés selon le modèle de couplage $\mathcal{M}$ choisi (ligne 3) qui vise à maximiser le nombre d'intervalles communs possible. Le choix de l'algorithme de couplage est donné à l'utilisateur. Nous utiliserons l'heuristique de couplage nommée $\texttt{IILCS}_{\mathcal{M}}$ [7] comme base pour la suite de nos travaux. Ensuite, chaque couple de gènes est renommé et les gènes non couplés sont retirés par $\mathcal{M}$*-élagage* afin d'obtenir deux génomes corrigés qui soient deux permutations (ligne 4).

À partir de ces deux permutations, le calcul des intervalles communs est réalisé (ligne 5) en utilisant l'algorithme proposé dans [102].

5.2.2 L'heuristique $\texttt{IILCS}_{\mathcal{M}}$

Le choix de l'algorithme de couplage (ligne 3 de l'algorithme 8) est l'étape la plus délicate de `Match&Watch`. Angibaud *et al.* [7] ont proposé une heuristique, nommée $\texttt{IILCS}_{\mathcal{M}}$ (Improved Iteration Longest Common Substring) afin de calculer, étant donné un modèle de couplage $\mathcal{M}$, le nombre maximum d'intervalles communs à deux génomes de manière approchée. L'heuristique $\texttt{IILCS}_{\mathcal{M}}$ utilise la notion de sous-chaîne commune.

Définition 34 (Sous-chaîne commune)**.** *Soient un alphabet $\mathcal{F}$, G_0 et G_1 deux chaînes définies sur $\mathcal{F}$. Une sous-chaîne commune à G_0 et G_1 est une chaîne qui est à la fois sous-chaîne de G_0 et sous-chaîne de G_1. Par exemple, soient l'alphabet $\mathcal{F} = \{1, 2, 3, 4\}$ et les chaînes $G_0 = 1\ 2\ 1\ 3\ 4\ 1$ et $G_1 = 2\ 3\ 4\ 1\ 2\ 1$. La chaîne $3\ 4\ 1$ est une sous-chaîne commune à G_0 et G_1.*

L'algorithme 9 présente $\texttt{IILCS}_{Max}$, la variante de $\texttt{IILCS}_{\mathcal{M}}$ pour le modèle de couplage maximum. Il génère un nombre d'intervalles communs très proche du nombre maximum. En effet sur 40 comparaisons de paires de génomes de γ-protobactéries, $\texttt{IILCS}_{Max}$ donne en moyenne un nombre d'intervalles à 98,9% du nombre maximum [8], en atteignant souvent le nombre maximum. La variante $\texttt{IILCS}_{Ex}$, pour le modèle de couplage exemplaire, doit veiller à ce qu'il n'y ait qu'un seul couplage par famille de gène. Pour cela, si deux gènes de la même famille apparaissent dans une sous-chaîne commune, nous ne couplons que la première occurrence d'un gène g et nous supprimons, une fois le couplage fait, toutes les autres occurrences de la famille $|g|$ dans les deux génomes.

Nous noterons $\texttt{M\&W-IILCS}_{\mathcal{M}}$ la variante de `Match&Watch` qui utilise l'heuristique de couplage $\texttt{IILCS}_{\mathcal{M}}$.

5.2.3 Intervalles communs maximaux

Le nombre d'intervalles communs étant très important (plus de 10000 sur chaque expérimentation réalisée), Angibaud *et al.* [4] proposent de sélectionner une partie d'entre eux. En partant de la constatation qu'un intervalle commun peut contenir d'autres intervalles communs, il semble intéressant de ne conserver que les plus grands. La procédure suivante permet donc de filtrer les intervalles obtenus afin de conserver les plus "intéressants". Angibaud *et al.* les appellent intervalles communs *maximaux*. Nous partons de $\mathcal{I}$ l'ensemble de tous les intervalles communs et nous suivons, dans l'ordre, les étapes suivantes pour obtenir $\mathcal{I}_{max}$, l'ensemble des intervalles communs maximaux :

Algorithme 9 Heuristique $\texttt{IILCS}_{Max}$

ENTRÉES: Deux génomes G_0 et G_1
SORTIES: Un couplage de gènes entre G_0 et G_1 qui vise à maximiser le nombre d'intervalles communs.

1: Soient G'_0 une copie de G_0 et G'_1 une copie de G_1.
2: **Tantque** Il existe un gène pouvant être couplé **faire**
3: Calculer une plus longue sous-chaîne S commune (à une inversion complète près) aux deux génomes et ne contenant aucun gène couplé.
4: Coupler les gènes de S de manière naturelle (i.e. soit dans le sens de lecture, soit dans le sens inverse de lecture)
5: Retirer de G'_0 (resp. G'_1) les gènes non couplés pour lesquels il ne reste aucun gène non couplé de la même famille dans G'_1 (resp. G'_0)
6: **Fin tantque**
7: Retourner le couplage de gènes obtenu.

1. Soit $\mathcal{I}_{max}$ une copie de $\mathcal{I}$.
2. Chaque intervalle qui contient tous les gènes d'un des deux génomes est retiré de $\mathcal{I}_{max}$. En effet, le génome entier est un intervalle commun trivial qui n'est pas informatif en génomique comparative.
3. Si nous travaillons sur les génomes circulaires, à chaque intervalle commun I dans $\mathcal{I}_{max}$ il existe un intervalle commun complémentaire I' dans $\mathcal{I}_{max}$. Dans ce cas là, nous considérons le plus petit des deux comme le plus informatif d'un point de vue biologique, ce qui signifie que nous retirons de $\mathcal{I}_{max}$ le plus grand des deux.
4. Nous retirons aussi de $\mathcal{I}_{max}$ chaque intervalle commun qui contient seulement un unique gène, car un gène seul n'est pas très informatif.
5. De même nous retirons de $\mathcal{I}_{max}$ chaque intervalle qui est inclus dans un autre intervalle de $\mathcal{I}_{max}$. Le $\mathcal{I}_{max}$ résultant forme maintenant l'ensemble des intervalles communs *maximaux*.

5.3 L'ajout de données *omiques* : vers une explication des processus biologiques conservés

Afin, de comprendre les raisons de la conservation, ensemble, des gènes d'un intervalle commun, nous avons décidé d'intégrer de la connaissance supplémentaire au calcul de ces intervalles. Ainsi, nous espérons pouvoir fournir une explication sur la conservation des intervalles de gènes, comme par exemple : les gènes de cet intervalle encodent des enzymes qui catalysent les réactions d'une même chaîne de réactions, ou bien les gènes d'un même intervalle sont coexprimés dans un contexte donné... Nous avons choisi de prendre en compte ce type d'informations métaboliques ou d'expression de gènes dans notre travail.

5.3.1 De $\texttt{IILCS}_{\mathcal{M}}$ à $\texttt{IISCS}_{\mathcal{M}}$

Nous proposons de raffiner les intervalles communs en utilisant de l'information *omique* connue sur l'une des deux espèces comparées via leur génome. Nous appellerons cette espèce *l'espèce de référence*, et son génome, le *génome de référence*. Sans perte de généralité, nous supposerons que le génome de référence est noté G_0. À chaque intervalle de gènes I du génome G_0, nous associons un score dit *d'intérêt*, qui décrit l'intérêt de l'intervalle I au regard des propriétés *omiques* étudiées dans l'organisme de

Algorithme 10 L'heuristique $\texttt{IISCS}_{Max}$

ENTRÉES: Deux génomes G_0 et G_1,
une matrice de scores A de taille $|G_0| \times |G_0|$

SORTIES: Un couplage de gènes entre G_0 et G_1 qui vise à favoriser les intervalles communs de meilleur score.

1: Soient G'_0 une copie de G_0 et G'_1 une copie de G_1.
2: **Tantque** il existe un gène pouvant être couplé **faire**
3: *Prendre la sous-chaîne S commune (à une inversion complète près) aux deux génomes telle que S ne contient aucun gène couplé et soit de meilleur score dans la matrice A. S'il existe plusieurs sous-chaînes candidates, en choisir une au hasard.*
4: Coupler les gènes de S de manière naturelle (i.e. soit dans le sens de lecture, soit dans le sens inverse de lecture)
5: Retirer de G'_0 (resp. G'_1) les gènes non couplés pour lesquels il ne reste aucun gène non couplé de la même famille dans G'_1 (resp. G'_0)
6: **Fin tantque**
7: Retourner le couplage de gènes obtenu.

référence. Plus ce score sera important, plus l'intervalle I sera intéressant. Nous constituons ainsi une matrice A, nommée *matrice d'intérêt*, qui indique, pour tous les intervalles de gènes $[g_i, g_j]$ entre les positions i et j dans G_0, le score d'intérêt $A[i, j]$ correspondant. Dans le cas des génomes composés de chromosomes circulaires, les intervalles $[g_i, g_j]$ et $[g_j, g_i]$ sont distincts, et n'ont pas nécessairement le même score d'intérêt. L'intervalle $[g_i, g_j]$ décrit l'intervalle de gènes allant du gène g_i au gène g_j en suivant le sens du brin principal du chromosome.

La prise en compte de ces informations supplémentaires intervient au niveau du calcul du couplage dans `Match&Watch`. Nous avons pour objectif de définir un ordre de couplage des gènes en sélectionnant en priorité les sous-chaînes de gènes les plus intéressantes. Il est cependant aussi possible d'envisager d'autres objectifs comme par exemple optimiser le score global du couplage. Afin de réaliser cette opération, pour un modèle de couplage $\mathcal{M}$ donné, nous avons modifié l'heuristique $\texttt{IILCS}_{\mathcal{M}}$ et ainsi créé une nouvelle heuristique nommée $\texttt{IISCS}_{\mathcal{M}}$ (Improved Iteration *Scoring* Common Substring). L'Algorithme 10 présente cette heuristique pour la variante $\texttt{IISCS}_{Max}$ (Algorithme 9) qui utilise le modèle de couplage maximum. La différence entre $\texttt{IILCS}_{\mathcal{M}}$ et $\texttt{IISCS}_{\mathcal{M}}$ tient à l'ajout de la matrice A en entrée, et la modification de la ligne 3 de l'algorithme $\texttt{IILCS}_{Max}$, qui consiste à prendre la séquence de gènes meilleur score plutôt que la plus longue séquence. Le score de chaque séquence est le score d'intérêt de l'intervalle minimum de gènes induit par cette séquence. Nous n'obtenons plus ainsi un couplage de gènes qui tente de maximiser le nombre d'intervalles communs, mais un couplage de gènes qui tente de favoriser les intervalles communs de meilleur score. Nous noterons $\texttt{M\&W-IISCS}_{\mathcal{M}}$ la variante de `Match&Watch` qui utilise l'heuristique de couplage $\texttt{IISCS}_{\mathcal{M}}$. Ainsi $\texttt{M\&W-IISCS}_{Max}$ indiquera, que nous utilisons `Match&Watch` avec l'heuristique $\texttt{IISCS}_{Max}$ pour le modèle de couplage maximum.

La variante $\texttt{IISCS}_{Ex}$, pour le modèle de couplage exemplaire, suit la même modification de $\texttt{IILCS}_{Ex}$ à partir de $\texttt{IILCS}_{Ex}$. La modification doit veiller à ce qu'il n'y ait qu'un seul couplage par famille de gène. Pour cela, si deux gènes de la même famille apparaissent dans une sous-chaîne commune, nous ne couplons que la première occurrence d'un gène g et nous supprimons, une fois le couplage fait, toutes les autres occurrences de la famille $|g|$ dans les deux génomes.

L'heuristique $\texttt{IISCS}_{\mathcal{M}}$ est une généralisation de $\texttt{IILCS}_{\mathcal{M}}$. En effet, la matrice A de la Figure 5.2 encode, pour chaque intervalle de gènes allant de g_i à g_j, la longueur de l'intervalle de gènes sur le

$$A = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ n & 1 & 2 & \cdots & n-1 \\ n-1 & n & 1 & \cdots & n-2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 3 & 4 & \cdots & 1 \end{pmatrix}$$

Figure 5.2 – Reproduction de l'heuristique $\texttt{IILCS}_{\mathcal{M}}$ à partir de l'heuristique $\texttt{IISCS}_{\mathcal{M}}$. La matrice A permet cette reproduction car la valeur de chaque cellule $A[i, j]$ est la longueur de l'intervalle de gènes qui commence avec le gène à la position i sur le génome et qui finit avec le gène à la position j.

génome de référence. Les intervalles de gènes les plus longs seront ceux de plus grand score, et ils seront sélectionnés d'abord. Il est aussi tout à fait possible de prendre en compte n'importe quelle autre information supplémentaire grâce à la matrice A que nous avons introduite lors de la génération du couplage des gènes permettant le calcul des intervalles communs.

5.3.2 Nouvelles données prises en compte

Nous avons réalisé deux expérimentations de calcul d'intervalles communs, l'une intégrant de l'information métabolique, et l'autre intégrant de l'information de corrélation d'expression de gènes. Nous avons comparé les résultats obtenus à ceux de l'expérimentation proposée dans Angibaud *et al.* [6], qui consiste dans le calcul des intervalles sans tenir compte de données *omiques*. Nous pourrons ainsi évaluer par la suite l'apport de l'intégration de données *omiques* dans le calcul des intervalles communs.

5.3.2.1 Prendre en compte l'information métabolique

Pour prendre en compte l'information métabolique dans le calcul des intervalles communs, nous avons intégré, pour l'organisme de G_0, les aspects génomiques et métaboliques dans $\mathcal{G}_{int}$ afin de générer une matrice d'intérêt A_m. Dans le chapitre précédent, nous avons constaté que les mesures de coefficient de voisinage $\overline{w}_d$ et que la densité génomique d_G (voir section 3.3.2) étaient intéressantes pour discriminer les k-SIPs de $\mathcal{G}_{col}$ contenant des opérons, groupes de gènes contigus qui dépendent d'un même facteur de transcription. La densité génomique, en particulier, est une mesure très intéressante. Lorsque la densité d'un k-SIP tend vers 1, sa valeur maximale, cela indique que l'ensemble des gènes du k-SIP sont proches sur le génome. Par conséquent, les gènes du k-SIP induisent un intervalle de gènes sur le génome encodant pour un enchaînement de réactions. Ces intervalles sont porteurs d'information métabolique significative comme nous l'avons étudié dans le chapitre précédant. Nous avons donc utilisé ceux-ci pour guider le choix du couplage des gènes de deux génomes distincts lors de la recherche des intervalles communs avec `Match&Watch`.

Travail préparatoire

Nous avons généré un instance de $\mathcal{G}_{int}$ en intégrant le génome de référence G_0 et le réseau métabolique associé à l'organisme référence. Nous avons pondéré $\mathcal{G}_{int}$ avec la distance de colocalisation (voir section 3.3.5).

Le problème du chemin qui maximise la densité génomique

Nous nous sommes intéressés, pour chaque couple de gènes (g_s, g_t) du génome G_0, à la recherche dans $\mathcal{G}_{int}$ d'un $srdr$-chemin qui maximise la densité génomique :

Définition 35 (Le problème du $srdr$-chemin de densité maximum **MAXGDP**). *Soit $\mathcal{G}_{int}$= (S, A), et g_s et g_t deux gènes du génome G_0. Le problème du $srdr$-*chemin de densité maximum *consiste à trouver un $srdr$-chemin $ch = x_1x_2 \dots x_n$ dans $\mathcal{G}_{int}$ avec $x_1 = (g_s, r)$, $x_n = (g_t, r')$ et $r \neq r'$ tel que $d_G(ch)$ soit maximum.*

Nous allons d'abord étudier la complexité de **MAXGDP** en montrant qu'un de ses sous-problème simple, nommée **MAXDEP**, est NP-complet. Nous proposerons ensuite une heuristique qui calcule un chemin de densité maximum, et finalement nous expliquerons comment générer la matrice d'intérêt métabolique A_m à partir d'un ensemble des chemins de densité maximum entre chaque couple de gènes.

Le sous-problème MAXDEP

Le sous-problème **MAXDEP** (maximum density elementary path) est le problème **MAXGDP** réduit à la classe d'instances particulières de $\mathcal{G}_{int}$= (S, A) où (a) le génome intégré est constitué d'un unique chromosome linéaire, (b) chaque gène est couplé exactement à une réaction, et (c) chaque réaction est couplée exactement à un gène. Autrement dit pour tous sommets $x_i = (g, r)$ et $x_j = (g', r')$ avec $g \neq g'$, nous avons $r \neq r'$, et vice versa. Le point (a) nous permet de définir un ordre total sur l'ensemble des gènes du génome : leur ordre d'apparition selon le sens de lecture du chromosome. Le point (b), quant à lui, nous autorise à renommer chaque sommet $x = (g, r)$ de $\mathcal{G}_{int}$ par le gène g. En renommant chaque gène du génome par sa position sur le génome, nous obtenons le génome $G_0 = 1\ 2\ \dots\ n$ et nous pouvons ainsi considérer $\mathcal{G}_{int}$ comme un graphe orienté dont chaque sommet est étiqueté par un entier de 1 à n. Le point (c) garantit, pour toute réaction, qu'un chemin passera au plus une fois par cette réaction. La recherche d'un $srdr$-chemin consiste donc à rechercher un chemin élémentaire, c'est-à-dire un chemin sans circuit.

Ainsi nous pouvons écrire le problème d'optimisation **MAXDEP** de la façon suivante :

Définition 36 (Le problème d'optimisation **MAXDEP**). *Soient $H = (V, E)$ un graphe orienté avec $V = \{1, 2, \dots, n\}$, s et t deux sommets de H. Le problème* maximum density elementary path *consiste à trouver le chemin élémentaire $ch = x_1x_2 \dots x_p$ allant de $s = x_1$ à $t = x_p$ dans H qui maximise $d_G(ch)$ avec :*

$$d_G(ch) = \frac{p}{\max_{1 \leq i \leq p}\{x_i\} - \min_{1 \leq j \leq p}\{x_j\} + 1}$$

Afin de montrer que **MAXDEP** est NP-complet, nous allons étudier la complexité du problème de décision α-**DEP** qui lui est associé et qui est défini comme suit :

Définition 37 (Le problème de décision α-**DEP**). *Soit $H = (V, E)$ un graphe orienté avec $V = \{1, 2, \dots, n\}$, et soient s et t deux sommets de H et $\alpha \in \mathbb{R}$ avec $0 \leq \alpha \leq 1$. Existe-t-il un chemin élémentaire $ch = x_1x_2 \dots x_p$ allant de $s = x_1$ à $t = x_p$ dans H tel que $d_G(ch) \geq \alpha$?*

$$\textit{avec } d_G(ch) = \frac{p}{\max_{1 \leq i \leq p}\{x_i\} - \min_{1 \leq j \leq p}\{x_j\} + 1}$$

Montrons que α-DEP $\in$ NP. Pour un chemin élémentaire ch dans G, la vérification $d_G(ch) \geq \alpha$ s'effectue en un temps linéaire en fonction du nombre de sommets du chemin. Ainsi, le problème de décision α-**DEP** est dans NP.

Montrons que α-DEP est NP-dur Pour démontrer que α-**DEP** est NP-dur, nous allons réduire le problème du plus long chemin à α-**DEP**.

Définition 38 (Le problème de décision du plus long chemin élémentaire)**.** *Soient un graphe orienté $H = (V, E)$ avec $V = \{1, 2, \ldots, n\}$, un entier k, s et t deux sommets de H. Existe-t-il un chemin élémentaire $ch = x_1 x_2 \ldots x_{p+1}$ de $s = x_1$ à $t = x_{p+1}$ dans H tel que $p \geq k$?*

Avec les notations des définitions 37 et 38, nous proposons le théorème suivant :

Théorème 2. *Soit $H' = (V', E')$ le graphe orienté défini par $V' = V \cup \{0, n+1\}$ et $E' = E \cup \{0s, t(n+1)\}$ et soit $\alpha = \frac{k+3}{n+2}$. Il existe existe un chemin élémentaire de longueur supérieure ou égale à k entre s et t dans H si et seulement s'il existe un chemin élémentaire ch' de 0 à $n+1$ dans H' tel que $d_G(ch') \geq \alpha$.*

Démonstration. Notons que dans H', tout chemin élémentaire $ch' = x_1 x_2 \ldots x_{l+1}$ de longueur l entre 0 et $n+1$ satisfait $x_2 = s$ et $x_l = t$. Par construction de H', nous avons $\max_{1 \leq i \leq l+1}(x_i) = n+1$ et $\min_{1 \leq i \leq l+1}(x_i) = 0$, ce qui entraîne $d_G(ch') = \frac{l+1}{n+2}$. Ainsi, nous obtenons la suite d'équivalences qui suit :

$$\begin{aligned} d_G(ch') \geq \alpha &\Leftrightarrow \frac{l+1}{n+2} \geq \frac{k+3}{n+2} \\ &\Leftrightarrow l+1 \geq k+3 \\ &\Leftrightarrow l-2 \geq k \end{aligned}$$

Ce qui est équivalent à : $x_2 \ldots x_l$ est un chemin élémentaire de longueur supérieure ou égale à k. Ainsi, l'équivalence est démontrée. □

Par conséquent, le théorème 2 est valide, et nous montrons ainsi que le problème de décision α-**DEP** est NP-dur. Donc, le problème de décision de associé à **MAXGDP** (qui est clairement dans NP) est aussi NP-complet.

Algorithme de résolution de MAXDEP Le théorème 2 nous indique qu'il est possible de résoudre le problème du plus long chemin élémentaire en le transformant en une instance du problème **MAXDEP**. Il est aussi possible de résoudre n'importe quelle instance du problème **MAXDEP** en résolvant le problème du plus long chemin élémentaire. L'Algorithme 11 décrit la manière de faire.

Dans un graphe $H = (V, E)$ avec $V = \{1, 2, \ldots, n\}$ et s et t deux sommets distincts donnés, l'Algorithme 11 cherche ch_{max} un chemin de s à t tel que $d_G(ch_{max})$ est maximum. La démonstration suivante prouve que l'Algorithme 11 calcule bien ch_{max}.

Démonstration Afin de montrer que le chemin ch_{max} retourné par l'Algorithme 11 a la densité maximum, soit d_{max} la densité maximum atteignable et soit P_0 un chemin de s à t tel que $d_G(P_0) = d_{max}$.

Avec la notation x_0 (respectivement y_0) pour le sommet maximum (respectivement minimum) de P_0, la notation $||P_0||$ pour la longueur de P_0 (c'est-à-dire son nombre d'arcs), et la notation $I_{[x..y]}$ pour l'ensemble des sommets dans l'intervalle $[x..y]$, nous avons :

$$d_{max} = d_G(P_0) = \frac{||P_0|| + 1}{|I_{[x_0..y_0]}|}.$$

Maintenant, nous montrons que la valeur d_{max} est atteinte par le chemin ch choisi dans la ligne 5 de l'algorithme lorsque, aux lignes 2 et 3, les valeurs $x = x_0$ et respectivement $y = y_0$ sont choisies. Nous notons que, puisque P_0 est un chemin de s à t, nous avons $x_0 \leq min(s,t)$ et $y_0 \geq max(s,t)$.

Soit donc ch le chemin trouvé à la ligne 5, et soient x' (respectivement y') le sommet maximum (respectivement minimum) de ch. Alors $x \leq x'$ et $y' \leq y$, puisque ch est un chemin dans H', et donc $[x'..y'] \subseteq [x_0..y_0]$. De plus, $||ch|| \geq ||P_0||$ puisque ch est un plus long chemin dans H'.

Trois possibilités existent pour ch :

1. $||ch|| > ||P_0||$. Si ceci est vrai, alors

$$d_G(ch) = \frac{||ch|| + 1}{|I_{[x'..y']}|} \geq \frac{||ch|| + 1}{|I_{[x_0..y_0]}|} > \frac{||P_0|| + 1}{|I_{[x_0..y_0]}|} = d_G(P_0) = d_{max}.$$

   Mais alors la maximalité de d_{max} est contredite.

2. $||ch|| = ||P_0||$ et $[x'..y'] \subset [x_0..y_0]$. Si ceci est vrai, alors

$$d_G(ch) = \frac{||ch|| + 1}{|I_{[x'..y']}|} = \frac{||P_0|| + 1}{|I_{[x'..y']}|} > \frac{||P_0|| + 1}{|I_{[x_0..y_0]}|} = d_G(P_0) = d_{max}.$$

   Encore une fois, la maximalité de d_{max} est contredite.

3. $||ch|| = ||P_0||$ et $[x'..y'] = [x_0..y_0]$. Dans ce cas, nous avons évidement $d_G(ch) = d_G(P_0) = d_{max}$.

Le seul cas possible parmi les cas énumérés est le cas 3. Par conséquent, à la ligne 8 de l'algorithme le chemin ch_{max} est un chemin de densité d_{max}. Il ne sera plus modifié par la suite, et il sera donc retourné par l'algorithme à la ligne 13.

Le problème du calcul exact du plus long chemin élémentaire est un problème NP-complet. Cependant, il existe un algorithme d'approximation qui fournit une solution à ce problème avec une précision de l'ordre $O(n/\log n)$ par rapport à la solution exacte [2]. Nous pouvons élaborer un algorithme non exact qui résout le problème **MAXDEP** en remplaçant la version exacte de l'algorithme du plus long chemin par sa version approchée.

Algorithme d'approximation pour résoudre MAXDGP

Le sous-problème **MAXDEP** ne concerne que certaines instances de **MAXDGP** qui sont caractérisées par un génome linéaire G_0 dans $\mathcal{G}_{int}$, dont les gènes ne catalysent qu'au plus une seule réaction via leurs enzymes, i.e. chaque gène intervient dans au plus un seul sommet de $\mathcal{G}_{int}$, et dont les réaction ne sont pas catalysées par plus d'un gène. En pratique, nous travaillons sur des génomes circulaires où la notion d'ordre total des gènes ne tient plus, où certains gènes catalysent plusieurs réactions via leurs enzymes et où certaines réactions sont catalysées par plusieurs gènes.

Une possibilité existe afin de prendre en compte les instances de $\mathcal{G}_{int}$ où le génome est circulaire, dont chaque gène ne catalyse qu'au plus une réaction, et dont chaque réaction est catalysée par au plus un gène. Soit $C = g_1\ g_2\ \ldots\ g_n$ le génome circulaire de l'instance $\mathcal{G}_C$ de $\mathcal{G}_{int}$. Lorsque nous cherchons un chemin de densité maximum entre g_s et g_t dans $\mathcal{G}_C$ nous procédons en suivant les étapes suivantes.

1. Nous transformons C en un génome linéaire $L = g'_1\ g'_2\ \ldots\ g'_{2n}$ avec $g'_i = g'_{i+n} = g_i$. Nous appelons g_i l'antécédent de g'_i et g'_{i+n}. Nous générons le graphe $\mathcal{G}_L$ en dédoublant chaque sommets (g_i, r) en deux sommet (g'_i, r) et (g'_{i+n}, r).

Algorithme 11 Algorithme `ComputeMAXDEP`

ENTRÉES: $H = (V, E)$ un graphe orienté avec $V = \{1, 2, \ldots, n\}$,
$s, t \in V$
SORTIES: un chemin ch_{max} dans H entre s et t maximisant $d_G(ch_{max})$.

1: $ch_{max} \leftarrow \epsilon$
2: **Pour** $x \leftarrow 1$ **à** $\min(s, t)$ **faire**
3: **Pour** $y \leftarrow \max(s, t)$ **à** n **faire**
4: Soit H' le sous-graphe de H induit par $I_{[x..y]}$
5: Chercher dans H' le plus long (en nombre d'arcs) chemin élémentaire ch entre s et t
6: **Si** ch existe **alors**
7: **Si** $d_G(ch_{max}) < d_G(ch)$ **alors**
8: $ch_{max} \leftarrow ch$
9: **Finsi**
10: **Finsi**
11: **Fin pour**
12: **Fin pour**
13: **Retourner** ch_{max}

2. Nous étiquetons ensuite dans $\mathcal{G}_L$ chaque sommet par la position de son gène dans L. Nous revenons ainsi dans un cas similaire à **MAXDEP**.
3. Nous supposons, sans perte de généralité, que $s < t$. Dans $\mathcal{G}_L$, nous cherchons u un chemin de densité maximum entre s et t et v un chemin de densité maximum entre t et $s + n$ de telle sorte que les longueurs de u et de v soient chacune inférieure ou égale à n. Nous pourrons utiliser l'algorithme `ComputeMAXDEP` (voir Algorithme 11) dans lequel nous limiterons la taille des intervalles $[x..y]$ à au plus n. Les chemins seront des $srdr$-chemins, car nous ne pouvons pas avoir, par construction, deux sommets de même réaction dans H' (ligne 4 de l'Algorithme 11) tant que la taille de l'intervalle $[x..y]$ est inférieure à n.
4. Le chemin w de plus grande densité génomique d_G entre u et v est le chemin de densité maximum dans $\mathcal{G}_L$. Il suffit de reporter le chemin w sur $\mathcal{G}_C$ à l'aide de la notion d'antécédent pour obtenir le chemin de densité maximum entre g_s et g_t dans $\mathcal{G}_C$. Sa densité d_G est la même que celle de w.

Le réétiquetage des sommets de $\mathcal{G}_{int}$ par la position des gènes sur le génome G_0 est le point clé pour la recherche d'un $srdr$-chemin de densité maximum dans $\mathcal{G}_{int}$ à l'aide du problème de la recherche du plus long chemin élémentaire. Dès lors qu'il existe des gènes qui catalysent plusieurs réactions via leurs enzymes, il devient impossible de renommer les sommets de $\mathcal{G}_{int}$ par la position des gènes eux-mêmes, car nous perdons la bijection entre un gène et un sommet du graphe (i.e. plusieurs sommets de $\mathcal{G}_{int}$ auraient la même étiquette). De plus, si nous tenons compte de la contrainte supplémentaire que le chemin doit être un $srdr$-chemin, les méthodes présentées précédemment deviennent difficilement exploitables.

Nous proposons donc l'heuristique `ComputeDensePath` (voir Algorithme 12), fortement inspirée de l'heuristique `ComputeSIPs` (voir Algorithme 6, page 40). Pour $\mathcal{G}_{int}$ et un k donné, cette heuristique calcule un ensemble de k plus courts $srdr$-chemins entre les sommets associés au gène g_s et ceux associés au gène g_t. Le chemin de cet ensemble qui à la plus forte densité génomique d_G sera sélectionné comme le chemin de densité génomique maximum. Plus le paramètre k est élevé, plus nous aurons de chances d'obtenir le véritable $srdr$-chemin de densité génomique maximum.

Algorithme 12 L'algorithme `ComputeDensePath`

ENTRÉES: $\mathcal{G}_{int}$ le modèle intégré ayant pour unités catalytiques les gènes d'un génome circulaire,
g_s un gène intial,
g_t un gène final,
k le nombre de plus courts $srdr$-chemins désirés.
SORTIES: un $srdr$-chemin entre g_s et g_t dans $\mathcal{G}_{int}$ qui vise à maximiser la densité génomique.
1: Soit H une copie de $\mathcal{G}_{int}$
2: Dans H, contracter tous les sommets (g_s, r) dans un unique sommet (g_s, R)
3: Dans le H résultant, contracter tous les sommets (g_t, r') dans un unique sommet (g_t, R')
4: Calculer $kpc \leftarrow$ `ComputeKSPATP`$(H, k, (g_s, R), (g_t, R'))$
5: Soit kpc_{int} l'ensemble des k chemins induits dans $\mathcal{G}_{int}$ par kpc
6: Soit ch un chemin de kpc_{int} tel que $d_G(ch) = max\{d_G(ch'), ch' \in kpc_{int}\}$
7: **Retourner** ch

Génération de la matrice d'intérêt A_m

Nous définissons $\mathcal{GIP}$ l'ensemble de tous les $srdr$-chemins de densité maximum obtenus entre chaque couple de gènes possible dans l'organisme de référence. Soit ch le $srdr$-chemin de densité maximum allant de g_i à g_j. Le score d'intérêt $A_m[i, j]$ n'est pas $d_G(ch)$, mais celui qui suit :

$$A_m[i,j] = \max_{ch \in \mathcal{GIP}} \{\textit{Jaccard}(E_{[g_i..g_j]}, E_{ch})\}$$

avec $E_{[g_i..g_j]}$ l'ensemble des gènes dans l'intervalle allant de g_i à g_j
et E_{ch} l'ensemble des gènes participant à ch

Ce choix a été privilégié car un $srdr$-chemin de densité maximum peut faire intervenir un intervalle plus grand que $[g_i..\ g_j]$ dans le calcul de sa densité comme le montre la Figure 5.3.

La mesure de Jaccard étant une mesure de similarité, lorsque nous comparons $E_{[g_i..g_j]}$ à E_{ch} pour chaque chemin ch de $\mathcal{GIP}$, nous quantifions à quel point I_{ch}, l'intervalle induit par ch, ressemble à $[g_i..g_j]$. Ainsi, lorsque la valeur de $\textit{Jaccard}(E_{[g_i..g_j]}, E_{ch})$ tend vers 1, ch est est très dense. Lorsque $\textit{Jaccard}(E_{[g_i..g_j]}, E_{ch})$ tend vers 0, alors $[g_i..g_j]$ et ch partagent peu des gènes. Cela décrit deux possibilités. Soit $d_G(ch)$ est faible ce qui ne le rend pas très intéressant ; soit $d_G(ch)$ est important ce qui implique qu'il existe un autre intervalle de gènes avec lequel ch aura une forte similarité, ce qui le rend intéressant sur un autre intervalle de gènes. Par conséquent, plus le score de $A_m[i, j]$ est important, plus l'intervalle $[g_i..g_j]$ sera significatif au regard de l'information métabolique, puisqu'il qualifiera un intervalle de gènes dont une grande partie est responsable d'un enchaînement de réactions.

5.3.2.2 Prendre en compte la corrélation d'expression des gènes

Nous avons également pris en compte les informations d'expression de gènes. Nous avons vu dans les chapitres précédents, en particulier à la section 2.3.2, que la mesure de la corrélation d'expression de deux gènes est une information importante pour définir si ces gènes sont fonctionnellement liés. Cependant, nous travaillons sur des intervalles composés de plusieurs gènes, et une comparaison deux à deux des gènes n'est pas adaptée. Nous avons donc défini une mesure d'intérêt, nommée *score de coexpression de l'intervalle de gène*, portant non pas sur deux, mais sur un ensemble de gènes. Nous avons tout d'abord

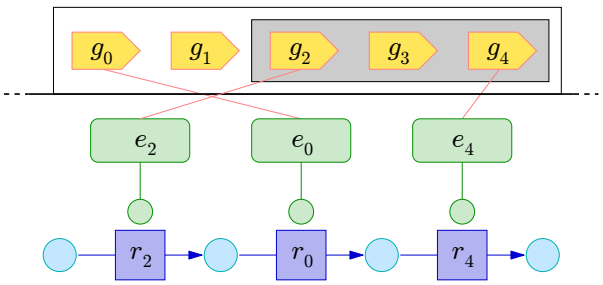


Figure 5.3 – Comparaison entre l'intervalle de gènes $[g_2..g_4]$ donné et l'intervalle induit par un $srdr$-chemin allant de g_2 à g_4. Le $srdr$-chemin ch allant de g_2 à g_4 est présenté ici en vue éclatée, avec les gènes (flèches épaisses) projetés sur le génome, les réactions (carrés) projetées sur le réseau métabolique formant une chaîne alternant réactions et métabolites (cercles), et les enzymes (rectangles arrondis) liant chaque gène dont elles sont le produit aux réactions qu'elles catalysent. L'intervalle de gènes $[g_2..g_4]$ est quant à lui représenté par le cadre gris sur le génome. Nous constatons que l'intervalle de gènes $[g_0..g_4]$ (cadre blanc) induit par ch "déborde" de $[g_2..g_4]$. En effet, ch utilise le gène g_0 qui n'appartient pas à $[g_2..g_4]$. La densité génomique de ch est $d_G(ch) = \frac{3}{5}$ et la mesure de Jaccard entre les gènes de ch et ceux de $[g_2..g_4]$ est égale à $0,5$.

calculé, pour chaque couple de gènes g_1 et g_2, le score de similarité d'expression $s(g_1, g_2)$ comme étant le coefficient de corrélation de Pearson entre la série de relevés des niveaux d'expression de g_1 et celle de g_2 (voir section 2.3.2 pour plus de détails). Nous obtenons ainsi une table de coexpression décrivant la similarité d'expression de chaque couple de gènes. Lorsque g_1 ou g_2 ne disposent pas de relevé, la similarité entre g_1 et g_2 est fixée à 0.

Soit g_i (resp. g_j) le gène à la position i (resp. j) dans le génome de référence G_0. Le score de coexpression de l'intervalle de gènes $[g_i..g_j]$ est alors défini comme la moyenne de tous les scores de similarité d'expression des couples de gènes de l'intervalle. Quand un gène n'a pas de niveau d'expression mesuré dans l'expérimentation, le gène n'est pas pris en compte dans le score de l'intervalle. C'est ce score que nous utilisons afin de générer la matrice d'intérêt A_c, définie comme suit :

$$A_c[i,j] = \frac{2 \cdot \sum_{i \leq x < y \leq j} |s(g_x, g_y)|}{n(n-1)}$$

où n est le nombre de gènes ayant une mesure d'expression, et $s(g_x, g_y)$ est la similarité d'expression entre les gènes g_x et g_y.

Si aucun gène de l'intervalle ne possède de valeur d'expression, le score de l'intervalle est fixé à 0, qui est le score par défaut. La corrélation d'expression d'un gène avec lui-même n'est pas prise en compte dans le score de coexpression de l'intervalle. Cela signifie, dans un intervalle constitué de n gènes dont les niveaux d'expression ont été mesurés, qu'il y a au plus $\sum_{i=1}^{n-1} i$ mesures de similarité d'expression, soit $\frac{n(n-1)}{2}$ mesures.

5.4 Évaluation des intervalles communs

Nous allons évaluer l'intérêt des intervalles communs obtenus sur *E. coli* et *V. Cholera* en utilisant les différents couplages de gènes dans `Match&Watch`. Nous avons utilisé des mesures qui diffèrent un peu de celles de la section 4.4. En effet, nous ne travaillons plus sur des k-SIPs, mais sur des intervalles

de gènes. Nous nous sommes en particulier intéressés à l'information opéronique contenue dans les intervalles communs, puisque les opérons, comme les intervalles communs, sont des groupes de gènes contigus.

Détection d'opérons

Pour quantifier l'intérêt opéronique des intervalles communs, nous avons comparé les intervalles communs de gènes avec les opérons. Pour mémoire, un opéron est une unité de transcription qui contient au moins deux gènes. Nous avons obtenu ces opérons de la base Ecocyc [59] pour *E. coli*. Contrairement au résultats présentés dans le chapitre 4, les opérons étudiés ici ne sont pas uniquement les opérons métaboliques, mais la totalité des opérons.

Notion d'intervalles opéroniques

Nous dirons qu'un intervalle de gènes I est un *intervalle non opéronique* (NOI : Non Operonic Interval) quand il n'existe aucun opéron dont les gènes forment un sous-ensemble des gènes de I. Au contraire, nous dirons qu'un ensemble de gènes est un *intervalle partiellement opéronique* (POI : Partially Operonic Interval) quand il existe une collection non-vide d'opérons dont l'ensemble des gènes est un sous-ensemble des gènes de l'intervalle, et un intervalle est un *intervalle pleinement opéronique* (FOI : Full Operonic Interval) quand il existe une collection non-vide d'opérons dont l'ensemble des gènes est exactement l'ensemble des gènes de l'intervalle. En utilisant ces qualiticatifs, nous pouvons dire qu'un intervalle de gènes *contient au moins un opéron* quand il est POI ou FOI, et que l'intervalle *ne contient aucun opéron* quand il est NOI.

Comparaison avec le hasard

Nous allons aussi vérifier si l'ensemble des intervalles communs maximaux obtenus est plus significatif d'un point de vue opéronique que n'importe quel autre ensemble d'intervalles de gènes. Nous avons donc mesuré, dans l'ensemble des intervalles communs maximaux obtenu via $\texttt{M\&W-IILCS}_{Max}$, la proportion d'intervalles qui sont FOI, POI et NOI. Nous avons comparé ces proportions avec les moyennes de celles obtenues sur 100 ensembles d'intervalles aléatoires. Chaque ensemble d'intervalles aléatoires a été constitué de la façon suivante : soit IC_{Max} l'ensemble des intervalles communs maximaux, n_x le nombre d'intervalles communs de longueur x et X l'ensemble des longueurs des différents intervalles de IC_{Max}. L'ensemble IC_{Rand} des intervalles aléatoires est obtenu en effectuant un tirage aléatoire dans le génome G_0 de n_x intervalles de longueur x pour chaque $x \in X$. Ainsi IC_{Rand} contient le même nombre d'intervalles de même longueur que IC_{Max}, ce qui le rend plus facilement comparable à IC_{Max}.

5.5 Résultats biologiques

5.5.1 Génomes investigués

Afin de tester notre méthode, nous avons comparé deux génomes de protobactéries : celui d'*Escherichia coli* K12 MG1655 [16] (un chromosome circulaire, identifiant NCBI : NC_000913) et celui de *Vibrio cholerae* 01 biovar eltor str. N16961 [49] (deux chromosomes circulaires, identifiants NCBI : NC_002505 & NC_002506). Nous utiliserons le génome d'*E. coli* comme le génome de référence (i.e. G_0), car la connaissance existante à propos d'*E. coli* est très importante par rapport à celle de *V. cholerae*.

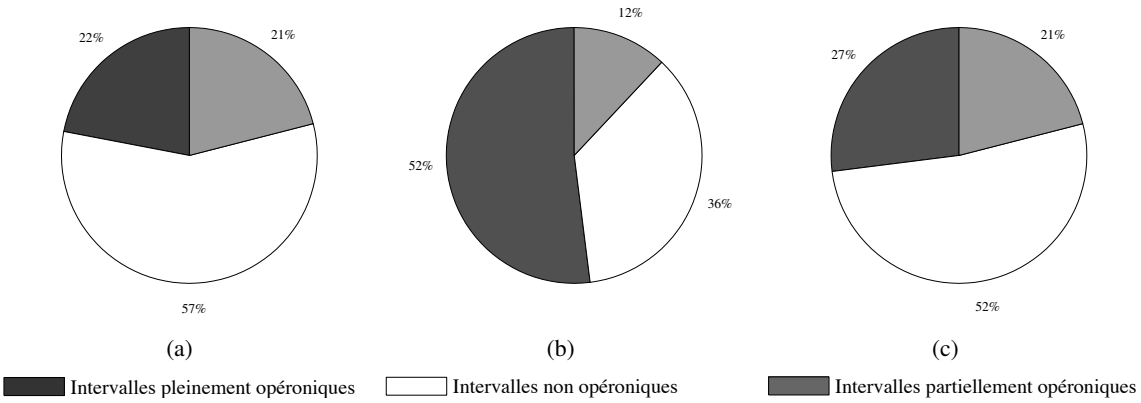


Figure 5.4 – Intérêt opéronique des intervalles communs. Proportion d'intervalles communs qui sont (1) pleinement opéroniques (FOI), (2) non opéroniques (NOI), et (3) partiellement opéroniques (FOI) pour (a) tous les intervalles communs, (b) les intervalles communs maximaux, et (c) les intervalles communs aléatoires.

5.5.2 Intervalles communs et intervalles communs maximaux

Dans notre application, nous avons choisi le modèle *couplage maximum* pour le couplage des gènes lors du calcul des intervalles communs. En effet, il permet de conserver un maximum de gènes communs aux deux génomes comparés.

Utilisation de M&W−IILCS$_{Max}$

Les résultats que nous présentons ici ont été présentés par Angibaud *et al.* dans [4]. Le calcul des intervalles à l'aide de M&W−IILCS$_{Max}$ (voir section 5.2.2) génère 11504 intervalles communs. Parmi ceux-ci, 325 constituent l'ensemble des intervalles communs maximaux. L'intérêt opéronique de ces intervalles communs maximaux est résumé dans la Figure 5.4. En effet, alors que seuls 43% des intervalles communs parmi tous les intervalles calculés contiennent au moins un opéron (FOI + POI), 64% des intervalles communs maximaux contiennent au moins un opéron. De même, lorsque 100 ensembles aléatoires d'intervalles (de taille similaire à celle de obtenue par l'ensemble des intervalles commun maximaux) sont générés, seuls 48%, en moyenne, des intervalles contiennent au moins un opéron. Ces résultats confirment donc bien l'intérêt des intervalles communs maximaux d'un point de vue opéronique et fonctionnel (puisqu'un opéron est une unité fonctionnelle du génome). Les intervalles communs maximaux sont donc de bons candidats pour évaluer l'intérêt d'intégrer des données *omiques* dans la comparaison de génomes en utilisant l'heuristique IISCS$_{\mathcal{M}}$.

Utilisation de M&W−IISCS$_{Max}$ avec la matrice d'intérêt A_m

Dans le cas de l'intégration d'informations métaboliques venant de *E. coli*, nous avons calculé l'ensemble $\mathcal{GIP}$ en utilisant le génome d'*E. coli* issu de la base NCBI (identifiant NC_000913) et le réseau métabolique d'*E. coli* issu de la base de données de voies métaboliques KEGG PATHWAYS (identifiant *eco*). Seuls 779 gènes parmi 4242 (18% du génome) codent pour une enzyme qui catalyse une réaction. L'ensemble $\mathcal{GIP}$ obtenu est composé de 333714 $srdr$-chemins, et nous permet de générer la matrice d'intérêt A_m. Le calcul des intervalles à l'aide de M&W−IISCS$_{Max}$ et la matrice A_m, que nous

nommons alors M&W−IISCS$_{Max}(A_m)$, génère 11468 intervalles communs. Ces intervalles communs diffèrent très peu de ceux obtenus en utilisant M&W−IILCS$_{Max}$. 318 d'entre eux forment l'ensemble des intervalles communs maximaux, qui diffère très peu de l'ensemble des intervalles communs maximaux obtenus en utilisant M&W−IILCS$_{Max}$.

Utilisation de M&W−IISCS$_{Max}$ avec la matrice d'intérêt A_c

Dans le cas de l'intégration de donnés d'expression de gènes, nous avons également utilisé les informations connues à propos de *E. coli*. Les informations d'expression proviennent de la base Gene Expression Omnibus (GEO) [36]. Nous avons utilisé plus précisément les informations venant de l'expérimentation GDS2580. Elle correspond à une série de relevés temporels décrivant la reprise de croissance d'une population bactérienne d'*E. coli* après une phase stationnaire [1], sur un medium LB (Lysogeny Broth - bouillon de lysogène) enrichi en glucose, qui est un medium riche en nutriments et donc propice à la croissance d'une population de bactéries. Dans cette expérience, 3623 gènes parmi 4242 (85% du génome) ont une mesure d'expression permettant de générer la matrice d'intérêt A_c. Le calcul des intervalles à l'aide de M&W−IISCS$_{Max}$ et la matrice A_c, que nous nommons alors M&W−IISCS$_{Max}(A_c)$, génère 11494 intervalles communs. Encore une fois, les intervalles communs obtenus ici diffèrent très peu de ceux obtenus en utilisant M&W−IILCS$_{Max}$. L'ensemble des intervalles communs maximaux est quant à lui constitué de 317 intervalles communs, et est très semblable à l'ensemble des intervalles communs maximaux obtenus en utilisant M&W−IILCS$_{Max}$.

L'information *omique* supplémentaire, qui prend place dans la matrice d'intérêt, ne semble pas faire varier beaucoup les intervalles communs obtenus lors des différentes expériences effectuées. Ceci peut s'expliquer par le fait que les génomes bactériens possèdent très peu de gènes dupliqués, limitant ainsi le choix des couplages de gènes possibles, et par là même la diversité des intervalles communs obtenus. L'information *omique* supplémentaire apporte néanmoins de l'information supplémentaire sur la raison pour laquelle les gènes constituant chaque intervalle commun sont conservés ensemble.

5.5.3 Résultats en fonction des informations *omiques* intégrées

Malgré une similarité évidente entre les résultats obtenus par M&W−IILCS$_{Max}$ et M&W−IISCS$_{Max}$, la méthode M&W−IISCS$_{Max}$ a le gros avantage d'associer un score (métabolique ou de coexpression) à chaque intervalle commun. Cela nous donne l'opportunité de sélectionner les intervalles les plus intéressants en nous basant sur la connaissance *omique*. Nous avons considéré que les intervalles communs maximaux intéressants sont ceux qui ont un score, dit *omique*, au dessus de 0,6 dans les matrices d'intérêt A_m ou A_c. Ainsi, 30 intervalles de gènes ont été sélectionnés dans le cas de l'intégration de données métaboliques, tandis que 86 l'ont été dans le cas de l'intégration de données de coexpression de gènes.

La Figure 5.5 présente ces résultats. Le raffinement obtenu par la sélection des intervalles communs maximaux en utilisant l'heuristique de couplage IISCS$_{Max}$ dans Match&Watch est évident : de 64% des intervalles contenant au moins un opéron (FOI + POI), à respectivement 90% et 89% en tenant compte de l'information métabolique ou de coexpression. En proportion, nous obtenons moins de NOI quand les données *omiques* sont prises en compte. Les intervalles communs conservés consistent donc, pour une majorité d'entre eux, en des opérons conservés au fil de l'évolution, en particulier quand les gènes de ces opérons sont responsables d'un même processus métabolique (lorsque l'intervalle a un score $\geq 0,6$ dans A_m), ou que leur expression est corrélée (lorsque l'intervalle a un score $\geq 0,6$ dans A_c).

1. Phase où la population bactérienne reste stable

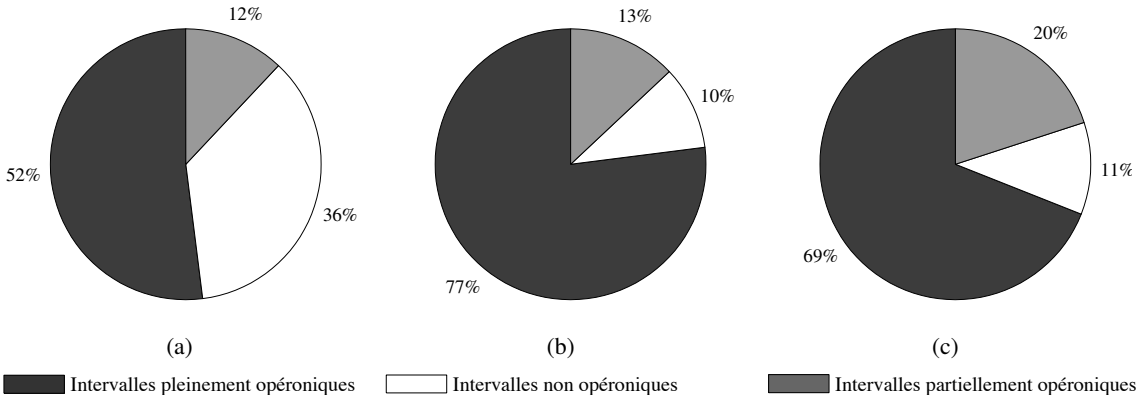


Figure 5.5 – Proportion d'intervalles communs maximaux pleinement, partiellement et non opéroniques pour chaque expérimentation d'intervalles communs maximaux entre *E. coli* et *V. cholerae* : intervalles communs maximaux obtenus par (a) M&W−IILCS$_{Max}$, (b) M&W−IISCS$_{Max}(A_m)$, intégrant de l'information métabolique, avec une limite de densité génomique fixée à 0,6, et (c) M&W−IISCS$_{Max}(A_c)$, intégrant les données d'expression de gènes de l'expérimentation GDS2580, avec une limite de score d'expression d'intervalle fixée à 0,6.

D'autre part, seuls 13 intervalles sélectionnés apparaissent à la fois en tenant compte des données métaboliques et des données de coexpression. Cela suggère que les gènes d'un intervalle qui participent à un enchaînement de réactions ne sont pas forcément corrélés au niveau de leur expression alors même qu'ils forment un ou plusieurs opérons. Parmi ces intervalles, 11 (84,62%) contiennent au moins un opéron (FOI + POI). Ils indiquent un ensemble de gènes successifs, responsables d'un enchaînement de réactions successives sur le réseau métabolique, et dont l'expression (des gènes) est corrélée dans un environnement propice à la croissance de la population bactérienne suite à une carence en sucres.

5.5.4 Prédiction d'opérons vs précision fonctionnelle

Les résultats de M&W−IISCS$_{Max}$ semblent précis d'un point de vue fonctionnel pour comparer des génomes bactériens. Ainsi, il est tentant de passer de la signification des intervalles à la prédiction d'opérons. Nous avons utilisé les deux mesures [91] évaluant la qualité de la prédiction d'opérons. La première est la *sensibilité*, qui représente le taux de paires de gènes consécutifs à l'intérieur d'un opéron qui sont correctement prédites. Cela veut dire que plus la sensibilité est élevée, plus la technique est efficace pour reconnaître la composition en gènes des opérons. La seconde est la *spécificité*, qui présente le taux de paires de gènes en bordure d'opéron (un gène dedans et un gène dehors) correctement prédites. Plus la spécificité est élevée, plus la méthode est efficace pour borner les limites des opérons. Quand nous appliquons ces mesures sur les opérons d'*E. coli*, les intervalles communs maximaux obtenus par M&W−IILCS$_{Max}$ montrent une sensibilité de $45,47\%$ et une spécificité de $21,42\%$. En intégrant de l'information métabolique via M&W−IISCS$_{Max}(A_m)$ puis en ne sélectionnant que les intervalles maximaux ayant un score supérieur à $0,6$ dans A_m, nous obtenons une sensibilité de $30,26\%$ et une spécificité de $12,48\%$. De même, en intégrant de l'information de coexpression de gènes via M&W−IISCS$_{Max}(A_c)$ puis en filtrant ceux avec un score supérieur à $0,6$ dans A_c, nous obtenons une sensibilité de $18,75\%$ et une spécificité de $15,39\%$.

Les résultats obtenus indiquent que M&W−IISCS$_{Max}$ est mauvais dans la prédiction d'opérons. Cependant, 30 intervalles communs maximaux couvrent 50 opérons dans le cas de l'intégration de données

métaboliques. Cela implique que les intervalles communs maximaux contiennent une organisation plus complexe que seulement un unique opéron par intervalle et explique pourquoi les scores de spécificité sont faibles. Nous obtenons par exemple un intervalle composé de quatre opérons : *fepDGC*, *fepB*, *entS*, *entCEBA-ybdB*. Il est important de remarquer que ces quatres opérons sont co-régulés par des activateurs communs : Crp et Fur. De même, les produits des opérons *fepDGC* et *fepB* composent un unique transporteur ABC qui indique le lien fonctionnel et essentiel de ces deux opérons ensembles, corroborant leur sélection commune au sein d'un intervalle. De plus, l'opéron *entS* est divergeant et antagoniste à l'opéron *fepDGC*. Ces éléments génétiques observés appuient l'idée que ces opérons ont été conservés ensemble au fil de l'évolution par nécessité biologique. Cela suggère qu'il ne faut pas uniquement observer les opérons, mais aussi comment ceux-ci interagissent entre-eux et participent, partiellement ou totalement, à un même processus biologique, soit en étudiant les enchaînements de réactions dont ils sont responsables, soit en en étudiant la corrélation d'expression de leur gènes. De telles structures complexes ne peuvent pas être prédites en utilisant les mesures standard de prédiction d'opérons telles que la sensibilité et la spécificité, et expliquent d'ailleurs les faibles scores observés lors de l'évaluation de la prédiction d'opérons.

5.6 Conclusion

L'usage de données hétérogènes ensemble a déjà été exploitée dans la prédiction d'opérons par Chuang *et al.* [27]. Les auteurs ont utilisé en particulier les informations de distance métabolique, de distance intergénique, et les clusters de gènes orthologues (COG) afin de déterminer précisément les opérons présents dans un génome. Dans notre cas, nous avons proposé ici `M&W-IISCS`$_{Max}$, un algorithme de comparaison de génomes qui utilise, en plus des données génomiques, des données *omiques* supplémentaires (i.e. métaboliques et d'expression de gènes). Nous ne cherchons pas d'opérons en particulier à l'aide des informations prises en compte en plus des génomes, mais à fournir une explication sur la conservation, ensemble, de certains groupes de gènes contigus et identifiés par la notion d'intervalles communs. Les intervalles communs maximaux sont les plus intéressants, car beaucoup d'entre eux sont constitués d'un ou plusieurs opérons. Le fait que ces opérons soient conservés ensemble suggère un lien fonctionnel possible entre eux. Comme déjà montré dans [27], un tel ajout améliore significativement la compréhension fonctionnelle (et donc les annotations de gènes) des génomes comparés. L'intégration de l'information métabolique indique clairement que certains opérons sont liés car une grande partie de leurs gènes participe à des enchaînements de réactions. L'intégration de l'information de corrélation d'expression, quant à elle, indique que ces opérons sont liés car leurs gènes sont exprimés de façon similaire dans un contexte environnemental donné, et donc que ces gènes sont conservés car ils interviennent dans un même comportement bactérien.

CHAPITRE 6

Conclusion

Dans ce mémoire, nous nous sommes intéressés à l'analyse de systèmes bactériens en tenant compte de divers types d'informations *omiques*.

6.1 Travail effectué

Nous avons mis en place, dans un premier temps, une méthode exploratoire, nommée `SIPPER` (voir chapitre 3), qui fonctionne en trois étapes. Tout d'abord, (A) `SIPPER` consiste à générer le graphe orienté, appelé *modèle intégré* ou $\mathcal{G}_{int}$, d'une bactérie à partir de son réseau métabolique et son génome pour la version "gène", ou de son réseau métabolique et de l'ensemble de ses enzymes pour la version "enzyme". Les arcs de $\mathcal{G}_{int}$ sont pondérés par une distance génomique, transcriptomique ou enzymatique selon l'information *omique* prise en compte. Ensuite, (B) pour tout couple d'ensembles disjoints de sommets, nous avons généré un ensemble de k plus courts chemins sans répétitions de réactions, nommé k-SIP. Nous nous sommes particulièrement intéressés aux k-SIPs entre chaque couple de réactions, dont nous avons étudié par la suite l'intérêt biologique en fonction de la pondération de $\mathcal{G}_{int}$. Nous avons fait une étude en utilisant la distance de colocalisation (voir $\mathcal{G}_{col}$ dans la section 4.5), distance qui tient compte de la proximité des gènes sur le génome, et la distance de coexpression (voir $\mathcal{G}_{coexp}$ dans la section 4.6), qui tient compte de la corrélation d'expression des gènes entre eux. Enfin, (C) nous avons utilisé les critères de coefficient de voisinage $\overline{w}_d$, dépendant de la distance utilisée dans $\mathcal{G}_{int}$, et de densité génomique d_G, indépendante de la distance choisie, pour discriminer certains k-SIPs (voir chapitre 4). Il est apparu, en analysant les résultats obtenus, que ces mesures, en particulier celle de densité génomique, se prêtaient bien à l'identification d'opérons métaboliques, mais qu'elles n'étaient pas des plus adaptées pour la découverte de modules métaboliques.

Dans un second temps, nous avons mis en place une autre méthode, nommée `M&W-IISCS`$_{\mathcal{M}}$ (voir chapitre 5), qui consiste en la comparaison de deux génomes, mais en intégrant et en utilisant, par rapport aux approches classiques, de la connaissance *omique*. `M&W-IISCS`$_{\mathcal{M}}$ calcule, en tenant compte des scores d'intérêt dépendant d'une propriété *omique* étudiée, un ensemble des intervalles communs maximaux de plus grand intérêt. Nous avons réalisé deux comparaisons entre les génomes de *E. coli* et de *V. Cholerae* avec cette méthode. La première utilise l'information métabolique de *E. coli*, la seconde utilise la corrélation d'expression des gènes de *E. coli*. Nous avons pu expliquer les raisons de la conservation de certains intervalles communs maximaux, et montrer que les intervalles communs maximaux sont des outils intéressants pour la découverte de groupes d'opérons proches conservés ensemble durant l'évolution.

Grâce à ces différents travaux, nous avons pu remarquer plusieurs propriétés sur l'organisation du génome et du métabolisme bactérien d'*E. coli*.

6.2 Organisation du génome et du réseau métabolique.

Les résultats obtenus dans le chapitre 4, lors des deux applications de SIPPER sur *E. coli* (voir $\mathcal{G}_{col}$ et $\mathcal{G}_{coexp}$), suggèrent une nature modulaire cohérente du génome et du réseau métabolique de *E. coli*. En effet, nous y retrouvons des k-SIPs de forte densité génomique (i.e. $> 0,7$) correspondant à des opérons auxquels s'ajoutent quelques gènes voisins, mais aussi des k-SIPs correspondant à des modules de KEGG. Dans le cas des réseaux métaboliques, ce découpage modulaire a déjà été défendu dans [96, 103, 83] et dans le cas des génomes, chaque opéron constitue une entité fonctionnelle. Les modules métaboliques de KEGG incarnent eux-aussi des fonctions biologiques. L'analyse détaillée des modules présentée dans les Annexes B et C montre que les k-SIPs opéroniques et les k-SIPs modulaires sont assez distincts, même si certains gènes voisins appartiennent au même module. Ces remarques mettent en avant le fait que les modules métaboliques et les opérons, qui sont des entités biologiques fonctionnelles, décrivent des étapes distinctes du fonctionnement d'un système vivant. Le fait que nous observions, dans l'étude de $\mathcal{G}_{coexp}$, des k-SIPs de forte densité génomique (i.e. $d_G \geq 0,7$), qui ne sont pas nécessairement des k-SIPs opéroniques, et dont les gènes ont des niveaux d'expression fortement corrélés, indique qu'il existe encore d'autres entités biologiques fonctionnelles.

Nous observons également dans $\mathcal{G}_{col}$ qu'un cinquième des k-SIPs contiennent à la fois des modules et des opérons (voir la Figure 4.10, dans la section 4.5). Cela indique que certaines de ces entités biologiques ont une coordination potentielle. Les intervalles communs maximaux obtenus avec M&W-IISCS$_{\mathcal{M}}$ appuient d'ailleurs cette observation : certains intervalles communs regroupent des opérons, parfois avec des gènes additionnels, conservés ensemble pour des raisons évolutionnaires. Ces intervalles communs maximaux sont proches des über-opérons [26] dans leur définition, et suggèrent une organisation hiérarchique du génome. Le gène constitue l'élément atomique, les opérons incarnant un premier niveau hiérarchique, puis les intervalles communs maximaux entre espèces suggèrent un second niveau hiérarchique. L'ajout d'information *omique* dans le calcul et l'analyse des intervalles communs maximaux permet d'expliquer de façon automatique pour quelle(s) raison(s) les gènes à l'intérieur d'un intervalle sont conservés ensemble.

6.3 Perspectives

Voici quelque perspectives que nous jugeons intéressantes d'étudier dans le futur.

6.3.1 Variations de la notion de k-SIP

Dans ce manuscrit, nous nous sommes essentiellement intéressés à la notion de k-SIP, un ensemble de k plus courts chemins sans répétition de réactions entre deux ensembles de réactions. Nous avons cependant abordé des variantes de k-SIPs. Par exemple, la recherche de $srdr$-chemins minimisant $\overline{w}_d$ (section 3.3.4, page 41) qui ressemble beaucoup aux problèmes (a) de la recherche du chemin minimisant le poids moyen de ses arcs [104], (b) du *all-pairs minimum average weighted length path* [106], (c) du plus court chemin restreint [47, 81]. Nous avons également travaillé sur des instances simples de la recherche de $srdr$-chemins de densité génomique maximum dans la section 5.3.2.1 à la page 91.

Ces différentes variations de la notion de k-SIPs nécessitent encore des travaux. Les problèmes liés à leur recherche nécessitent une étude de complexité, et il n'existe pas toujours d'algorithme exact pour calculer ces variations.

6.3.2 Identification de k-SIPs d'intérêt

Dans les sections 4.5.2.1 et 4.6.2.1, nous avons observé que la quasi-totalité des k-SIPs ont une signification biologique dans la base de connaissance Ecocyc [59]. Nos études à l'aide le coefficient de voisinage $\overline{w}_d$ et la densité génomique d_G nous ont permis d'en sélectionner certains, mais il peut exister d'autres critères de sélection pour faire sortir de nouveaux k-SIPs ayant une signification particulière. Une recherche de ces nouveaux critères est une piste intéressante à explorer.

6.3.3 Identification des gènes essentiels dans $\mathcal{G}_{int}$

Dans les sections 4.5.2.2 et 4.6.2.2, nous avons essayé de mettre en avant les gènes essentiels en fonction de leur fréquence d'apparition dans chaque ensemble k-$\mathcal{SIP}$ obtenu. Les mesures de centralité [32] n'étant pas adaptées pour la recherche de gènes essentiels, nous pourrions utiliser adapter les notions de *minimal cut sets* [61], ou bien de 'load points' et 'choke points' [82], ou encore utiliser d'autre travaux à propos de la robustesse au sein des réseaux biologiques [60] pour identifier ces gènes essentiels.

6.3.4 Reconstruction de pathways

Dans le cadre de la reconstruction de voies métaboliques, Faust *et al.* [39] utilisent des algorithmes de recherche de k plus courts chemins. En effet, la reconstruction de voies métaboliques, dans cet article, est réalisée en reliant, dans le réseau métabolique, les métabolites à la frontière d'une voie métabolique. Nous nommons ces métabolites par le terme *sommets limites* dans le graphe représentant le réseau métabolique. La reconstruction d'une voie métabolique consiste donc à relier entre eux les sommets limites à l'aide d'un arbre de poids minimum ; les poids utilisés étant une mesure classique sur les graphes : nombre d'arcs et/ou de nœuds, degré des nœuds,... La recherche d'un tel arbre est une instance du problème de l'arbre de Steiner [35] qui est un problème NP-complet. La résolution de ce problème est habituellement approchée [39] en calculant d'abord, pour chaque couple de sommets limites s et t, les k plus courts chemins entre s et t ; puis en sélectionnant l'ensemble de chemins qui, dans la collection de ceux calculés, minimisent le poids de l'arbre contenant tous les sommets limites. Dans notre cas nous pourrions utiliser les k-SIPs pour reconstruire des voies métaboliques dites intégrées en travaillant non pas sur le réseau métabolique mais sur $\mathcal{G}_{int}$. Cela permettrait d'utiliser de nouvelles distances biologiquement significatives. Le point clé de ce travail tiendra très certainement dans le choix de la distance.

6.3.5 Modules et clusters de k-SIPs

Lors de l'application de `SIPPER` dans le chapitre 4, nous avons obtenu un grand nombre de k-SIPs, qui se recoupent souvent sur certains sommets de $\mathcal{G}_{int}$, et certains d'entre-eux sont même inclus dans d'autres. Nous avons également observé qu'une partie des k-SIPs de densité génomique $d_G > 0,7$ dans $\mathcal{G}_{col}$ et $\mathcal{G}_{coexp}$ caractérise des opérons. Kovács *et al.* [63] indiquent que 60% des opérons métaboliques sont colinéaires, c'est-à-dire que l'ordre de succession de chaque couple de gènes d'un opéron dans le génome est le même que l'ordre dans lequel les réactions qu'ils catalysent s'enchaînent. Nous pouvons alors supposer que les k-SIPs opéroniques sont composés de sous-chemins denses, du fait de la colinéarité. Cette remarque nous suggère l'idée suivante, afin de limiter le nombre de k-SIPs : regrouper les k-SIPs de densité génomique assez forte ($> 0,7$) qui partagent des sommets communs dans $\mathcal{G}_{int}$ au sein d'un même groupe que nous appellerions module de k-SIPs. Des résultats préliminaires de cette application, dans le cadre de notre étude du *bioleeching* du cuivre par *A. ferroxydans*, sont d'ailleurs encourageants dans la reconnaissance d'opérons comme le montre la Figure 6.1.

6.3.6 Nouvelles distances biologiques

Dans la méthode SIPPER, nous avons avons créé un modèle générique, qu'il est possible de paramétrer en fonction d'hypothèses biologiques. Nous n'avons cependant testé que certaines possibilités dans ce manuscrit. Beaucoup d'autres variations sont possibles. Tout d'abord, nous avons suggéré, dans la conclusion (section 4.7) du chapitre 4, de pondérer les arcs par une distance reposant sur plusieurs observations biologiques. Nous pouvons également envisager de pondérer chaque arc par plusieurs distances biologiquement significatives (une liste non exhaustive est présentée dans la section 3.3.5). La recherche de k-SIPs dans $\mathcal{G}_{int}$ devient alors une recherche de plus courts chemins multi-objectifs [41], qui est un problème NP complet lorsqu'il n'existe pas d'ordre particulier sur les objectifs. L'utilisation de plusieurs distances nous semble naturelle dans l'identification de modules métaboliques comme ceux proposés dans la base de données KEGG. En effet, un module est constitué manuellement à partir de plusieurs notions biologiques. Il suffit alors d'avoir une mesure, non nécessairement triviale, par notion biologique prise en compte.

6.3.7 Prise en compte d'un contexte métabolique

D'autre modifications que le changement de pondération de $\mathcal{G}_{int}$ sont possibles. Par exemple, il est admis que les gènes formant un opéron sont tous sur le même brin d'un chromosome, c'est-à-dire que les gènes d'un opéron ont le même signe (voir section 2.1, page 6). Dans le cadre de la recherche d'opérons, il suffit de supprimer de $\mathcal{G}_{int}$ chaque arc xy, avec $x = (g, r)$ et $y = (g', r')$ et tel quel le signe de g soit différent de celui de g' pour prendre en compte cette information supplémentaire. Dans notre cas, nous ne l'avons pas fait, car nous n'avons pas travaillé dans le but particulier de l'identification d'opérons. D'autres études [27, 22] le font avec de bons résultats. Il est aussi possible d'introduire un aspect dynamique dans $\mathcal{G}_{int}$. Ainsi, lors de l'étude de $\mathcal{G}_{coexp}$, les niveaux d'expressions de gènes utilisés pour pondérer les arcs sont obtenus dans une contexte expérimental précis. La bactérie subit un stress et les niveaux d'expression de gènes relevés illustrent son comportement en réponse à ce stress. Dans le cas d'une carence en certains métabolites dans le milieu dans lequel se trouve la bactérie, il serait opportun de tenir compte de cette absence de métabolites dans $\mathcal{G}_{int}$ afin de détecter l'enchaînement de réactions propres à la bactérie dans un contexte donné. Cette absence se traduit par un retrait de certains sommets de $\mathcal{G}_{int}$.

6.3.8 Nouveaux scores d'intérêts d'un intervalle commun

Tout comme nos perspectives de travaux sur les différentes distances et mesures dans SIPPER, travailler sur différents scores d'intérêt d'intervalles communs est une piste pour nos futurs travaux. Il serait également possible de tenir compte non seulement de l'information *omique* connue à propos d'un seul organisme, mais aussi de l'information *omique* connue à propos de chaque organisme comparé. Nous devrions alors travailler non pas avec une matrice d'intérêt, mais avec deux matrices. Le passage à plusieurs matrices d'intérêts est même envisageable, pour peu que nous tenions compte de plusieurs types d'information *omique* pour chaque organisme comparé.

6.3.9 Comparer des génomes avec beaucoup de dupliqués

Notre application sur les intervalles communs a souligné un point intéressant. Les intervalles communs, lorsque nous utilisons les heuristiques M&W-IILCS$_{\mathcal{M}}$ et M&W-IISCS$_{\mathcal{M}}$, ne sont pas très différents. La (non)-variation des intervalles semble dépendre du nombre de gènes dupliqués présents sur

les génomes étudiés. Une application sur des génomes possédant beaucoup de gènes dupliqués, comme ceux des plantes, peut être envisagée, en notant toutefois que nous passons des organismes procaryotes, reconnus comme les plus simples du vivant, aux organismes eucaryotes, qui sont beaucoup plus compliqués.

Les différents points abordés dans cette conclusion montrent que, bien qu'il y ait quelques résultats prometteurs, d'autre travaux sont nécessaires. Nous sommes en particulier intéressés par ceux qui portent sur les distances et les mesures afin que ceux-ci valident la notion de k-SIP qui ne sera pas nécessairement telle que présentée, mais sous une forme plus évoluée. Le but de cette démarche est que la notion de k-SIP constitue une brique de base dans l'étude des systèmes biologiques.

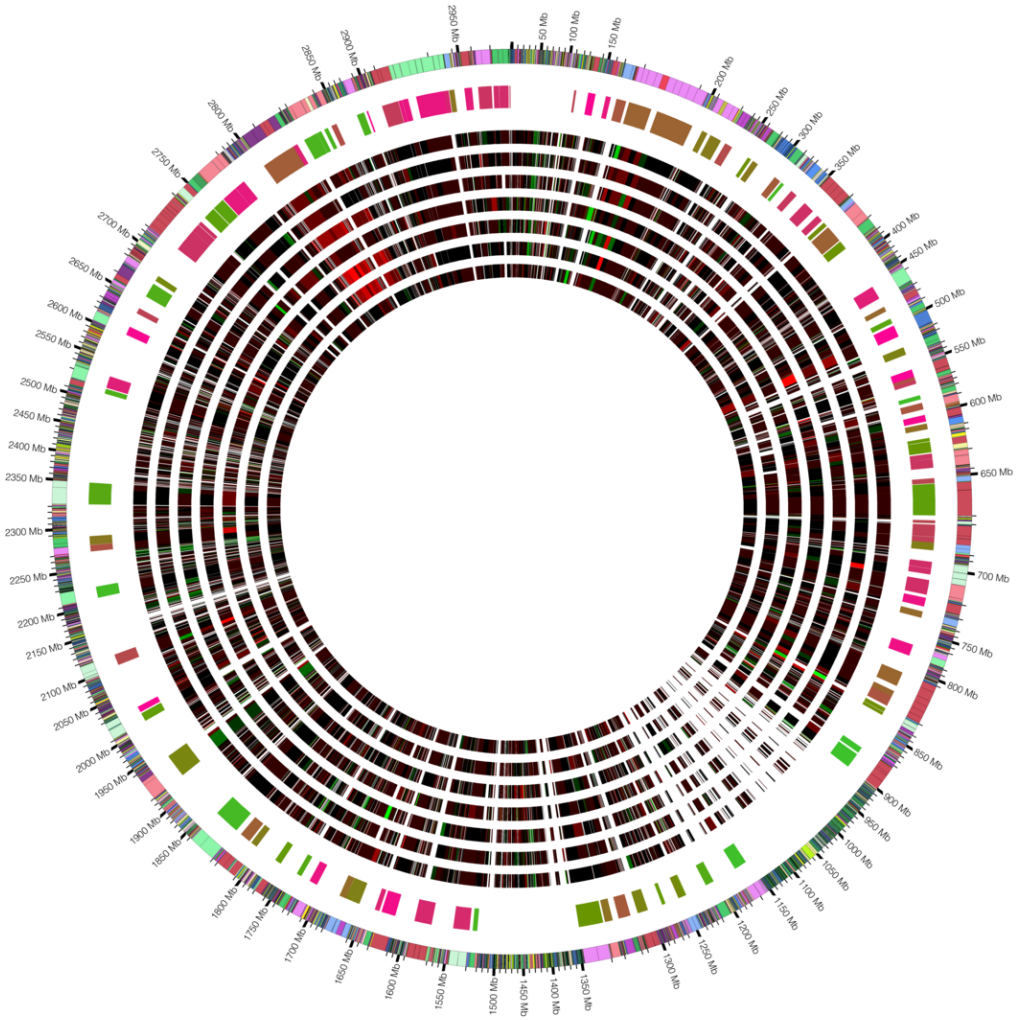


Figure 6.1 – Projection des modules de k-SIPs sur le génome de *A. ferroxydans*. Le cercle le plus externe représente les opérons sur le génome. Le second cercle en partant de l'extérieur représente les modules de k-SIPs, les autres cercles intérieurs présentent des profils d'expression des gènes, avec en vert les zones surexprimées dans le contexte expérimental, et en rouge les zones sous-exprimées. Les zones noires indiquent qu'il n'y a aucun changement d'expression.

Bibliographie

[1] B. ALBERTS : *Biologie moléculaire de la cellule*. Flammarion, 2004.

[2] N. ALON, R. YUSTER et U. ZWICK : Color-coding : a new method for finding simple paths, cycles and other small subgraphs within large graphs. *In Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, STOC '94, p. 326–335. ACM, 1994.

[3] S. F. ALTSCHUL, W. GISH, W. MILLER, E. W. MYERS et D. J. LIPMAN : Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[4] S. ANGIBAUD : *Comparaisons de génomes avec gènes dupliqués : étude théorique et algorithmes*. Thèse de doctorat, Université de Nantes, 2009.

[5] S. ANGIBAUD, P. BORDRON, D. EVEILLARD, G. FERTIN et I. RUSU : Integration of *omics* data to investigate common intervals. *In Proceedings of the 1st International Conference on Bioscience, Biochemistry and Bioinformatics (ICBBB 2011)*, p. 101–105, 2011.

[6] S. ANGIBAUD, D. EVEILLARD, G. FERTIN et I. RUSU : Comparing bacterial genomes by searching their common intervals. *In Proceedings of the 1st Bioinformatics and Computational Biology conference (BICoB 2009)*, vol. 5462 de *LNBI*, p. 102–113. Springer, 2009.

[7] S. ANGIBAUD, G. FERTIN, I. RUSU, A. THÉVENIN et S. VIALETTE : Efficient tools for computing the number of breakpoints and the number of adjacencies between two genomes with duplicate genes. *Journal of Computational Biology*, 15(8):1093–1115, 2008.

[8] S. ANGIBAUD, G. FERTIN, I. RUSU et S. VIALETTE : A pseudo-boolean general framework for computing rearrangement distances between genomes with duplicates. *Journal of Computational Biology*, 14(4):379–393, 2007.

[9] M. ARITA : 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, 2004.

[10] T. BABA, T. ARA, M. HASEGAWA, Y. TAKAI, Y. OKUMURA, M. BABA, K. A. DATSENKO, M. TOMITA, B. L. WANNER et H. MORI : Construction of *Escherichia coli K-12* in-frame, single-gene knockout mutants : the Keio collection. *Molecular Systems Biology*, 2(1):2006.0008, 2006.

[11] T. BARRETT, D. B. TROUP, S. E. WILHITE, P. LEDOUX, D. RUDNEV, C. EVANGELISTA, I. F. KIM, A. SOBOLEVA, M. TOMASHEVSKY, K. A. MARSHALL, K. H. PHILLIPPY, P. M. SHERMAN, R. N. MUERTTER et R. EDGAR : NCBI GEO : archive for high-throughput functional genomic data. *Nucleic Acids Research*, 37(Database issue):D885–890, 2009.

[12] Z. BARUTCUOGLU, R. E. SCHAPIRE et O. G. TROYANSKAYA : Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.

[13] O. BÉNICHOU, C. LOVERDO et R. VOITURIEZ : How gene colocalization can be optimized by tuning the diffusion constant of transcription factors. *Europhysics letters*, 84(3):1–3, 2008.

[14] C. BERGE : *Graphes et hypergraphes*. Dunod, 1973.

[15] A. BERGERON et J. STOYE : On the similarity of sets of permutations and its applications to genome comparison. *Journal of Computational Biology*, 13(7):1340–1354, 2006.

[16] F. R. BLATTNER, G. PLUNKETT, C. A. BLOCH, N. T. PERNA, V. BURLAND, M. RILEY, J. COLLADO-VIDES, J. D. GLASNER, C. K. RODE, G. F. MAYHEW, J. GREGOR, N. W. DAVIS, H. A. KIRKPATRICK, M. A. GOEDEN, D. J. ROSE, B. MAU et Y. SHAO : The complete genome sequence of *Escherichia coli K-12*. *Science*, 277(5331):1453–1462, 1997.

[17] P. BORDRON, D. EVEILLARD et I. RUSU : Integrated analysis of the gene neighboring impact on bacterial metabolic networks. *IET Systems Biology*, 5(4):261 – 268, 2011.

[18] P. BORDRON, D. EVEILLARD et I. RUSU : Sipper : A flexible method to integrate heterogeneous data into a metabolic network. *In Proceedings of the IEEE 1st International Conference on Computational Advances in Bio and Medical Sciences (ICCABS 2011)*, p. 40 – 45, 2011.

[19] P. E. BOURNE : The gene ontology's reference genome project : A unified framework for functional annotation across species. *PLoS Computational Biology*, 5(7):e1000431, 2009.

[20] F. BOYER, A. MORGAT, L. LABARRE, J. POTHIER et A. VIARI : Syntons, metabolons and interactons : an exact graph-theoretical approach for exploring neighbourhood between genomic and functional data. *Bioinformatics*, 21(23):4209–4215, 2005.

[21] E. I. BOYLE, S. WENG, J. GOLLUB, H. JIN, D. BOTSTEIN, J. M. CHERRY et G. SHERLOCK : Go : :termfinder–open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics*, 20(18):3710–3715, 2004.

[22] R. W. W. BROUWER, O. P. KUIPERS et S. A. F. T. van HIJUM : The relative value of operon predictions. *Briefings in Bioinformatics*, 9(5):367–375, 2008.

[23] M. R. J. CARLSON, B. ZHANG, Z. FANG, P. S. MISCHEL, S. HORVATH et S. F. NELSON : Gene connectivity, function, and sequence conservation : predictions from modular yeast co-expression networks. *BMC Genomics*, 7(1):40, 2006.

[24] R. CASPI, H. FOERSTER, C. A. FULCHER, P. KAIPA, M. KRUMMENACKER, M. LATENDRESSE, S. PALEY, S. Y. RHEE, A. G. SHEARER, C. TISSIER, T. C. WALK, P. ZHANG et P. D. KARP : The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Research*, 36(Database issue):D623–631, 2008.

[25] C. CHAUVE, G. FERTIN, R. RIZZI et S. VIALETTE : Genomes containing duplicates are hard to compare. *In Proceedings of the International Workshop on Bioinformatics Research and Applications (IWBRA 2006)*, vol. 3992. Springer Berlin Heidelberg, 2006.

[26] D. CHE, G. LI, F. MAO, H. WU et Y. XU : Detecting uber-operons in prokaryotic genomes. *Nucleic Acids Research*, 34(8):2418–2427, 2006.

[27] L.-Y. CHUANG, J.-H. TSAI et C.-H. YANG : Binary particle swarm optimization for operon prediction. *Nucleic Acids Research*, 38(12):e128, 2010.

[28] R. A. CLAYTON, O. WHITE, K. A. KETCHUM et J. C. VENTER : The first genome from the third domain of life. *Nature*, 387(6632):459–462, 1997.

[29] A. CORNISH-BOWDEN et M. L. CÁRDENAS : Information transfer in metabolic pathways. Effects of irreversible steps in computer models. *European Journal of Biochemistry*, 268(24):6616–6624, 2001.

[30] D. CROES, F. COUCHE, S. J. WODAK et J. van HELDEN : Metabolic pathfinding : inferring relevant pathways in biochemical networks. *Nucleic Acids Research*, 33(Web Server issue):W326–330, 2005.

[31] D. CROES, F. COUCHE, S. J. WODAK et J. van HELDEN : Inferring meaningful pathways in weighted metabolic networks. *Journal of Molecular Biology*, 356(1):222–236, 2006.

[32] G. del RIO, D. KOSCHÜTZKI et G. COELLO : How to identify essential genes from molecular networks ? *BMC Systems Biology*, 3(1):102, 2009.

[33] M. DEMEREC et P. E. HARTMAN : Complex loci in microorganisms. *Annual Review of Microbiology*, 13:377–406, 1959.

[34] E. W. DIJKSTRA : A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[35] C. DUIN, A. VOLGENANT et S. VOFL : Solving group Steiner problems as Steiner problems. *European Journal of Operational Research*, 154(1):323–329, 2004.

[36] R. EDGAR, M. DOMRACHEV et A. E. LASH : Gene Expression Omnibus : NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002.

[37] J. S. EDWARDS et B. Ø. PALSSON : Metabolic flux balance analysis and the in silico analysis of *Escherichia coli K-12* gene deletions. *BMC Bioinformatics*, 1(1):1, 2000.

[38] G. FANG, E. P. C. ROCHA et A. DANCHIN : How essential are nonessential genes ? *Molecular Biology and Evolution*, 22(11):2147–2156, 2005.

[39] K. FAUST, P. DUPONT, J. CALLUT et J. van HELDEN : Pathway discovery in metabolic networks by subgraph extraction. *Bioinformatics*, 26(9):1211–1218, 2010.

[40] G. FERTIN, A. LABARRE, I. RUSU, E. TANNIER et S. VIALETTE : *Combinatorics of genome rearrangements*. MIT Press, 2009.

[41] J. FIGUEIRA, S. GRECO et M. EHRGOTT : *Multiple Criteria Decision Analysis : State of the Art Surveys*. Springer, 2005.

[42] R. D. FLEISCHMANN, M. D. ADAMS, O. WHITE, R. A. CLAYTON, E. F. KIRKNESS, A. R. KERLAVAGE, C. J. BULT, J. F. TOMB, B. A. DOUGHERTY et J. M. MERRICK : Whole-genome random sequencing and assembly of *Haemophilus influenzae Rd. Science*, 269(5223):496–512, 1995.

[43] J. GAGNEUR, D. B. JACKSON et G. CASARI : Hierarchical analysis of dependency in metabolic networks. *Bioinformatics*, 19(8):1027–1034, 2003.

[44] M. Y. GALPERIN et E. V. KOONIN : Who's your neighbor ? New computational approaches for functional genomics. *Nature Biotechnology*, 18(6):609–613, 2000.

[45] A. L. P. GUEDES et L. MARKENZON : Directed Hypergraph Planarity. *Pesquisa Operacional*, 25(3):383–390, 2005.

[46] M. HASHIMOTO, T. ICHIMURA, H. MIZOGUCHI, K. TANAKA, K. FUJIMITSU, K. KEYAMURA, T. OTE, T. YAMAKAWA, Y. YAMAZAKI, H. MORI, T. KATAYAMA et J. ichi KATO : Cell size and nucleoid organization of engineered escherichia coli cells with a reduced genome. *Molecular Microbiology*, 55(1):137–149, 2005.

[47] R. HASSIN : Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1):36–42, 1992.

[48] M. HATTORI, Y. OKUNO, S. GOTO et M. KANEHISA : Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *Journal of the American Chemical Society*, 125(39):11853–11865, 2003.

[49] J. F. HEIDELBERG, J. A. EISEN, W. C. NELSON, R. A. CLAYTON, M. L. GWINN, R. J. DODSON, D. H. HAFT, E. K. HICKEY, J. D. PETERSON, L. UMAYAM, S. R. GILL, K. E. NELSON, T. D. READ, H. TETTELIN, D. RICHARDSON, M. D. ERMOLAEVA, J. VAMATHEVAN, S. BASS, H. QIN, I. DRAGOI, P. SELLERS, L. MCDONALD, T. UTTERBACK, R. D. FLEISHMANN, W. C. NIERMAN, O. WHITE, S. L. SALZBERG, H. O. SMITH, R. R. COLWELL, J. J. MEKALANOS, J. C. VENTER et C. M. FRASER : DNA sequence of both chromosomes of the cholera pathogen Vibrio cholerae. *Nature*, 406(6795):477–483, 2000.

[50] A. B. HORNE, T. C. HODGMAN, H. D. SPENCE et A. R. DALBY : Constructing an enzyme-centric view of metabolism. *Bioinformatics*, 20(13):2050–2055, 2004.

[51] N. ISHII, K. NAKAHIGASHI, T. BABA, M. ROBERT, T. SOGA, A. KANAI, T. HIRASAWA, M. NABA, K. HIRAI, A. HOQUE, P. Y. HO, Y. KAKAZU, K. SUGAWARA, S. IGARASHI, S. HARADA, T. MASUDA, N. SUGIYAMA, T. TOGASHI, M. HASEGAWA, Y. TAKAI, K. YUGI, K. ARAKAWA, N. IWATA, Y. TOYA, Y. NAKAYAMA, T. NISHIOKA, K. SHIMIZU, H. MORI et M. TOMITA : Multiple high-throughput analyses monitor the response of *E. coli* to perturbations. *Science*, 316(5824):593–597, 2007.

[52] F. JACOB, D. PERRIN, C. SÁNCHEZ et J. MONOD : L'opéron : groupe de gènes à expression coordonnée par un opérateur. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 250:1727–1729, 1960.

[53] H. JEONG, B. TOMBOR, R. ALBERT, Z. N. OLTVAI et A.-L. BARABÁSI : The large-scale organization of metabolic networks. *Nature*, 407(6804):651–4, 2000.

[54] A. R. JOYCE, J. L. REED, A. WHITE, R. EDWARDS, A. L. OSTERMAN, T. BABA, H. MORI, S. A. LESELY, B. Ø. PALSSON et S. AGARWALLA : Experimental and computational assessment of conditionally essential genes in *escherichia coli*. *Journal of Bacteriology*, 188(23):8259–8271, 2006.

[55] M. KANEHISA, M. ARAKI, S. GOTO, M. HATTORI, M. HIRAKAWA, M. ITOH, T. KATAYAMA, S. KAWASHIMA, S. OKUDA, T. TOKIMATSU et Y. YAMANISHI : KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, 36(Database issue):D480–484, 2008.

[56] M. KANEHISA et S. GOTO : KEGG : kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

[57] M. KANEHISA, S. GOTO, S. KAWASHIMA et A. NAKAYA : The KEGG databases at GenomeNet. *Nucleic Acids Research*, 30(1):42–46, 2002.

[58] J.-I. KATO et M. HASHIMOTO : Construction of consecutive deletions of the *Escherichia coli* chromosome. *Molecular Systems Biology*, 3(1), 2007.

[59] I. M. KESELER, C. BONAVIDES-MARTÍNEZ, J. COLLADO-VIDES, S. GAMA-CASTRO, R. P. GUNSALUS, D. A. JOHNSON, M. KRUMMENACKER, L. M. NOLAN, S. PALEY, I. T. PAULSEN, M. PERALTA-GIL, A. SANTOS-ZAVALETA, A. G. SHEARER et P. D. KARP : EcoCyc : a comprehensive view of *Escherichia coli* biology. *Nucleic Acids Research*, 37(Database issue):D464–470, 2009.

[60] H. KITANO : Biological robustness. *Nature Reviews Genetics*, 5(11):826–37, 2004.

[61] S. KLAMT et E. D. GILLES : Minimal cut sets in biochemical reaction networks. *Bioinformatics*, 20(2):226–234, 2004.

[62] P. KOLMAN et O. PANGRAC : On the complexity of paths avoiding forbidden pairs. *Discrete Applied Mathematics*, 157(13):2871–2876, 2009.

[63] K. KOVÁCS, L. D. HURST, B. PAPP et J. G. LAWRENCE : Stochasticity in protein levels drives colinearity of gene order in metabolic operons of *Escherichia coli*. *PLoS Biology*, 7(5):e1000115, 2009.

[64] N. C. KYRPIDES : Genomes OnLine Database (GOLD 1.0) : a monitor of complete and ongoing genome projects world-wide. *Bioinformatics*, 15(9):773–774, 1999.

[65] V. LACROIX, L. COTTRET, P. THÉBAULT et M.-F. SAGOT : An introduction to metabolic networks and their structural analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(4):594–617, 2008.

[66] A. LARHLIMI et A. BOCKMAYR : A new approach to flux coupling analysis of metabolic networks. *In* M. R. BERTHOLD, R. GLEN et I. FISCHER, éds : *Proceedings of the Computational Life Sciences II*, vol. 4216 de *Lecture Notes in Computer Science*, p. 205–215. Springer Berlin / Heidelberg, 2006.

[67] N. LEMKE, F. HERÉDIA, C. K. BARCELLOS, A. N. D. REIS et J. C. M. MOMBACH : Essentiality and damage in metabolic networks. *Bioinformatics*, 20(1):115–119, 2004.

[68] K. LIOLIOS, I.-M. A. CHEN, K. MAVROMATIS, N. TAVERNARAKIS, P. HUGENHOLTZ, V. M. MARKOWITZ et N. C. KYRPIDES : The Genomes On Line Database (GOLD) in 2009 : status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research*, 38(Database issue):D346–354, 2010.

[69] K. LIOLIOS, K. MAVROMATIS, N. TAVERNARAKIS et N. C. KYRPIDES : The Genomes On Line Database (GOLD) in 2007 : status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research*, 36(Database issue):D475–479, 2008.

[70] K. LIOLIOS, N. TAVERNARAKIS, P. HUGENHOLTZ et N. C. KYRPIDES : The Genomes On Line Database (GOLD) v.2 : a monitor of genome projects worldwide. *Nucleic Acids Research*, 34(Database issue):D332–334, 2006.

[71] H.-W. MA, X.-M. ZHAO, Y.-J. YUAN et A.-P. ZENG : Decomposition of metabolic network into functional modules based on the global connectivity structure of reaction graph. *Bioinformatics*, 20(12):1870–1876, 2004.

[72] S. MASLOV et K. SNEPPEN : Specificity and stability in topology of protein networks. *Science*, 296(5569):910–913, 2002.

[73] E. M. MCCREIGHT : A Space-Economical Suffix Tree Construction Algorithm. *Journal of the ACM*, 23(2):262–272, 1976.

[74] P. MICHALAK : Coexpression, coregulation, and cofunctionality of neighboring genes in eukaryotic genomes. *Genomics*, 91(3):243–248, 2008.

[75] H. J. MOTULSKY : *Biostatistique : une approche intuitive*. De Boeck Supérieur, 2002.

[76] A. NAKAYA, S. GOTO et M. KANEHISA : Extraction of correlated gene clusters by multiple graph comparison. *Genome Informatics*, 12:44–53, 2001.

[77] R. A. NOTEBAART, B. TEUSINK, R. J. SIEZEN et B. PAPP : Co-regulation of metabolic genes is better explained by flux coupling than by network distance. *PLoS Computational Biology*, 4(1):e26, 2008.

[78] H. OGATA, W. FUJIBUCHI, S. GOTO et M. KANEHISA : A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Research*, 28(20): 4021–4028, 2000.

[79] C. S. OSBORNE, L. CHAKALOVA, K. E. BROWN, D. CARTER, A. HORTON, E. DEBRAND, B. GOYENECHEA, J. A. MITCHELL, S. LOPES, W. REIK et P. FRASER : Active genes dynamically colocalize to shared sites of ongoing transcription. *Nature Genetics*, 36(10):1065–1071, 2004.

[80] J. A. PAPIN, J. STELLING, N. D. PRICE, S. KLAMT, S. SCHUSTER et B. Ø. PALSSON : Comparison of network-based pathway analysis methods. *Trends in Biotechnology*, 22(8):400–405, 2004.

[81] C. A. PHILLIPS : The network inhibition problem. *In Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC '93, p. 776–785. ACM, 1993.

[82] S. A. RAHMAN et D. SCHOMBURG : Observing local and global properties of metabolic pathways : 'load points' and 'choke points' in the metabolic networks. *Bioinformatics*, 22(14):1767–1774, 2006.

[83] E. RAVASZ, A. L. SOMERA, D. A. MONGRU, Z. N. OLTVAI et A.-L. BARABÁSI : Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, 2002.

[84] M. REMM, C. STROM et E. SONNHAMMER : Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of Molecular Biology*, 314:1041–1052, 2001.

[85] S. C. G. RISON, S. A. TEICHMANN et J. M. THORNTON : Homology, pathway distance and chromosomal localization of the small molecule metabolism enzymes in *Escherichia coli*. *Journal of Molecular Biology*, 318(3):911–932, 2002.

[86] I. RIVALS, L. PERSONNAZ, L. TAING et M. C. POTIER : Enrichment or depletion of a GO category within a class of genes : which test ? *Bioinformatics*, 23(4):401–407, 2007.

[87] E. P. C. ROCHA : The organization of the bacterial genome. *Annual Review of Genetics*, 42:211–233, 2008.

[88] J. RUAN, A. K. DEAN et W. ZHANG : A general co-expression network-based approach to gene expression analysis : comparison and applications. *BMC Systems Biology*, 4(1):8, 2010.

[89] T. L. M. A. A. S. J. Z. Z. Z. W. M. D. J. L. S F ALTSCHUL : Gapped BLAST and PSI-BLAST : a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[90] H. SALGADO, S. GAMA-CASTRO, M. PERALTA-GIL, E. DÍAZ-PEREDO, F. SÁNCHEZ-SOLANO, A. SANTOS-ZAVALETA, I. MARTÍNEZ-FLORES, V. JIMÉNEZ-JACINTO, C. BONAVIDES-MARTÍNEZ, J. SEGURA-SALAZAR, A. MARTÍNEZ-ANTONIO et J. COLLADO-VIDES : RegulonDB (version 5.0) : *Escherichia coli K-12* transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Research*, 34(Database issue):D394–397, 2006.

[91] H. SALGADO, G. MORENO-HAGELSIEB, T. SMITH et J. COLLADO-VIDES : Operons in *Escherichia coli* : genomic analyses and predictions. *Proceedings of the National Academy of Sciences of the United States of America*, 97(12):6652–6657, 2000.

[92] D. P. SANGURDEKAR, F. SRIENC et A. B. KHODURSKY : A classification based framework for quantitative description of large-scale microarray data. *Genome Biology*, 7(4):R32, 2006.

[93] D. SANKOFF : Genome rearrangement with gene families. *Bioinformatics*, 15(11):909–917, 1999.

[94] C. H. SCHILLING, S. SCHUSTER, B. Ø. PALSSON et R. HEINRICH : Metabolic pathway analysis : basic concepts and scientific applications in the post-genomic era. *Biotechnology Progress*, 15(3): 296–303, 1999.

[95] S. SCHUSTER, T. DANDEKAR et D. A. FELL : Detection of elementary flux modes in biochemical networks : a promising tool for pathway analysis and metabolic engineering. *Trends in Biotechnology*, 17(2):53–60, 1999.

[96] S. SCHUSTER, D. A. FELL et T. DANDEKAR : A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology*, 18(3):326–32, 2000.

[97] J.-M. SCHWARTZ, C. GAUGAIN, J. C. NACHER, A. D. DARUVAR et M. KANEHISA : Observing metabolic functions at the genome scale. *Genome Biology*, 8(6):R123, 2007.

[98] E. SIMEONIDIS, S. C. G. RISON, J. M. THORNTON, I. D. L. BOGLE et L. G. PAPAGEORGIOU : Analysis of metabolic networks using a pathway distance metric through linear programming. *Metabolic engineering*, 5(3):211–219, 2003.

[99] J. STELLING, S. KLAMT, K. BETTENBROCK, S. SCHUSTER et E. D. GILLES : Metabolic network structure determines key aspects of functionality and regulation. *Nature*, 420(6912):190–3, 2002.

[100] J. M. STUART, E. SEGAL, D. KOLLER et S. K. KIM : A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–255, 2003.

[101] J. TANG et B. M. E. MORET : Phylogenetic reconstruction from gene-rearrangement data with unequal gene content. *In Proceedings of the Workshop on Software Architectures for Dependable Systems*, vol. 2748 de *LNCS*, p. 37–46. Springer, 2003.

[102] T. UNO et M. YAGIURA : Fast Algorithms to Enumerate All Common Intervals of Two Permutations. *Algorithmica*, 26(2):290–309, 2000.

[103] A. WAGNER et D. A. FELL : The small world inside large metabolic networks. *Proceedings of the Royal Society - Biological Sciences*, 268(1478):1803–1810, 2001.

[104] S. WIMER, I. KOREN et I. CEDERBAUM : On paths with the shortest average arc length in weighted graphs. *Discrete Applied Mathematics*, 45(2):169–179, 1993.

[105] T. YAMADA, M. KANEHISA et S. GOTO : Extraction of phylogenetic network modules from the metabolic network. *BMC Bioinformatics*, 7(1):130, 2006.

[106] C. YANG : A pseudo-polynomial algorithm for detecting minimum weighted length paths in a network. *European Journal of Operational Research*, 57(1):123–131, 1992.

[107] Q. YANG et S.-H. SZE : Path matching and graph matching in biological networks. *Journal of Computational Biology*, 14(1):56–67, 2007.

[108] J. Y. YEN : Finding the k shortest loopless paths in a network. *Management Science*, 17:712–716, 1970.

[109] B. ZHANG et S. HORVATH : A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(1):Article17, 2005.

[110] Y. ZHENG, J. D. SZUSTAKOWSKI, L. FORTNOW, R. J. ROBERTS et S. KASIF : Computational identification of operons in microbial genomes. *Genome Research*, 12(8):1221–1230, 2002.

Liste des tableaux

Liste des figures

Liste des algorithmes

Table des matières

Index

Annexes

ANNEXE A

Exemple de déroulement d'un algorithme des k-plus courts chemins sans cycles

Afin de comprendre l'algorithme de Yen, pour la recherche des k plus courts chemins élémentaires, nous allons dérouler un exemple de recherche des 3 plus courts chemins entre 1 et 6 dans le graphe G de la Figure A.1

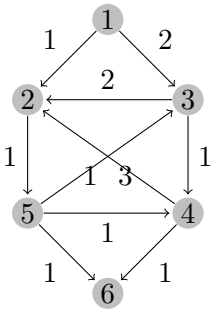


Figure A.1 – Un graphe orienté G sur lequel nous allons rechercher les 3 plus courts chemins de 1 à 6.

Dans le cadre de la recherche de chemins, lorsque plusieurs possibilités existent, nous utiliserons l'ordre lexicographique des sommets afin d'avoir un comportement déterministe dans l'exécution de l'algorithme de `Yen`.

Le déroulement de l'algorithme est disponible dans la Table A.1 dans les pages suivantes de cette annexe.

Table A.1 – Déroulement de la recherche des 3 plus courts chemins entre 1 et 6 dans G. Dans G, les sommets et arcs en vert appartiennent du plus court chemin allant du sommet en rouge jusqu'au sommet 6. Dans G et A, le sommet bleu correspond au point de déviation du chemin mis en évidence par ses arcs en en trait discontinu et le reste de l'arbre, les sommets et arcs en filigrane, sont les éléments qui ont été retirés de G en vue de la recherche des plus court sous chemins, le sommet rouge correspond au point de départ de la recherche du plus court sous-chemin jusqu'à 6.

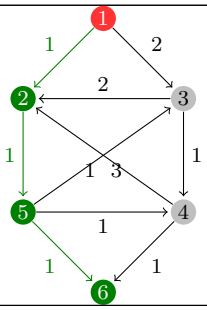
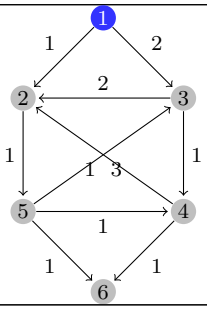
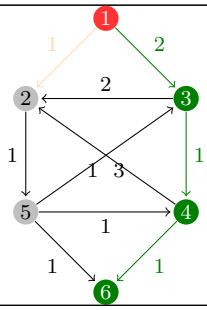
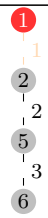
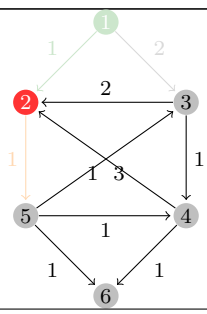
Ligne(s) de Algorithme 5	Graphe G	Arbre de préfixes communs maximaux A	*Chemins candidats*	*chemin étudié*	p	Commentaires
			Intialisation			
1-2		$\emptyset$	chemins | poids 1256 | 3			La première étape de l'algorithme consiste à rechercher le plus court chemin élémentaire entre 1 et 6 utilisant l'algorithme de `Dijkstra` (en vert dans G).
3			**1^{er} tour de la boucle tant que**			
4-8			chemins | poids	1256		Le plus court chemin candidat est 1256. Il est retiré des *Chemins candidats*, ajouté à A, et devient le *chemin étudié*. Le point de déviation de 1256 par rapport au reste de l'arbre est 1.
9			**1^{er} tour de la boucle pour**			
10-15			chemins | poids 1346 | 4	1256	1	L'arc 12 (en orange transparent) est retiré de G, et un nouveau plus court chemin entre 1 et 6 est calculé. Nous obtenons le chemin 1346 comme nouveau plus court chemin candidat et le préfixe p passe de ϵ à 1
16 -17			1 est retiré de G et retour à la ligne 9			
9			**2^{e} tour de la boucle pour**			
10-15			chemins | poids 1346 | 4	1256	12	L'arc 23 (en orange transparent) est retiré de G, et un nouveau plus court chemin entre 2 et 6 est recherché mais n'existe pas. Le préfixe p passe de 1 à 12

Table A.1 – suite de la page précédente

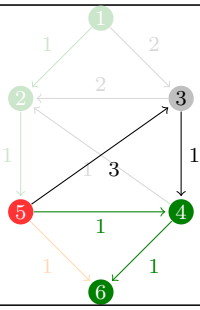
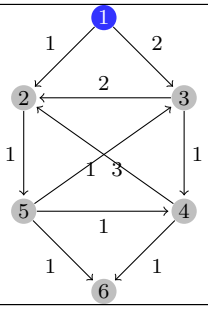
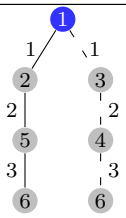
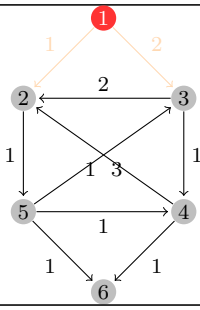
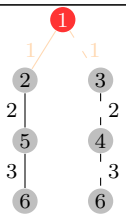
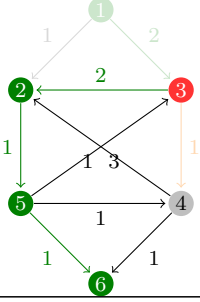
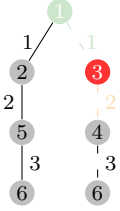
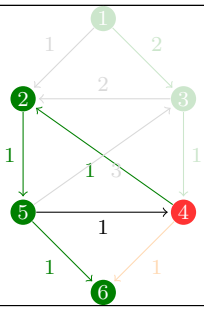
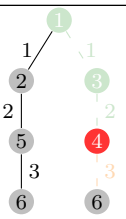
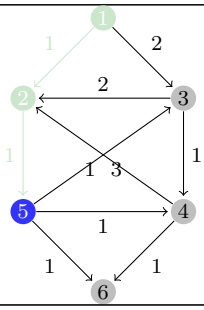
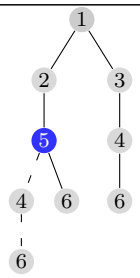
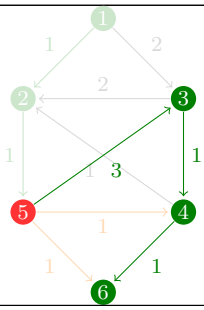
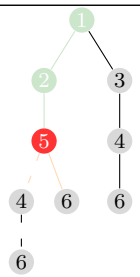
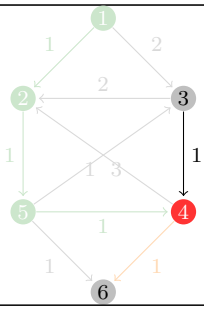
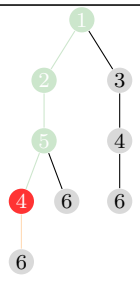
Ligne(s) de Algorithme 5	Graphe G	Arbre de préfixes communs maximaux A	*Chemins candidats*	*chemin étudié*	p	Commentaires
16 -17	2 est retiré de G et retour à la ligne 9					
9	**3^e tour de la boucle pour**					
10-15			chemins | poids 1346 | 4 12546 | 4	1256	125	L'arc 56 (en orange transparent) est retiré de G, et un nouveau plus court chemin entre 5 et 6 est recherché. Nous obtenons le chemin 12546 comme nouveau chemin candidat et le préfixe p passe de 12 à 125.
16 -17	5 est retiré de G et retour à la ligne 9					
9	Nous sommes à la fin de la boucle, nous allons ligne 18					
18-19	Remise des sommets et arcs retirés dans G et retour ligne 3					
3	**2^e tour de la boucle tant que**					
4-8			chemins | poids 12546 | 4	1346		Le plus court des chemins candidats est 1346. Il est retiré des *Chemins candidats*, ajouté à A, et devient le *chemin étudié*. Le point de déviation de 1346 par rapport au reste de l'arbre est 1.
9	**1^{er} tour de la boucle pour**					
10-15			chemins | poids 12546 | 4	1346	1	Les arcs 12 et 13 (en orange transparent) sont retirés de G, et un nouveau plus court chemin entre 1 et 6 est recherché, mais il n'en existe aucun. Le préfixe p passe de ϵ à 1
16 -17	1 est retiré de G et retour à la ligne 9					
9	**2^e tour de la boucle pour**					
10-15			chemins | poids 12546 | 4 13256 | 6	1346	13	L'arc 34 (en orange transparent) est retiré de G, et un nouveau plus court chemin entre 3 et 6 est recherché. Nous obtenons le chemin 13256 comme nouveau chemin candidat et le préfixe p passe de 1 à 13.
16 -17	3 est retiré de G et retour à la ligne 9					
9	**3^e tour de la boucle pour**					

Table A.1 – suite de la page précédente

Ligne(s) de Algorithme 5	Graphe G	Arbre de préfixes communs maximaux A	*Chemins candidats*	*chemin étudié*	p	Commentaires
10-15			chemins | poids 12546 | 4 13256 | 6 134256 | 6	1346	134	L'arc 46 (en orange transparent) est retiré de G, et un nouveau plus court chemin entre 4 et 6 est recherché. Nous obtenons le chemin 134256 comme nouveau chemin candidat et le préfixe p passe de 13 à 134.
16 -17	4 est retiré de G et retour à la ligne 9					
9	Nous sommes à la fin de la boucle, nous allons ligne 18					
18-19	Remise des sommets et arcs retirés dans G et retour ligne 3					
3	3^e **tour de la boucle tant que**					
4-8			chemins | poids 13256 | 6 134256 | 6	12546	12	Le plus court des chemins candidats est 12546. Il est retiré des *Chemins candidats*, ajouté à A, et devient le *chemin étudié*. Le point de déviation de 12546 par rapport au reste de l'arbre est 5, et le préfixe p est 12.
9	1^{er} **tour de la boucle pour**					
10-15			chemins | poids 13256 | 6 134256 | 6 125346 | 7	12546	125	Les arcs 54 et 56 (en orange transparent) sont retirés de G, et un nouveau plus court chemin entre 5 et 6 est recherché. Nous obtenons le chemin 125346 comme nouveau chemin candidat et le préfixe p passe de 12 à 125.
16 -17	5 est retiré de G et retour à la ligne 9					
9	2^e **tour de la boucle pour**					
10-15			chemins | poids 13256 | 6 134256 | 6 125346 | 7	12546	1254	L'arc 46 (en orange transparent) est retiré de G, et un nouveau plus court chemin entre 4 et 6 est recherché, mais il n'en existe aucun. Le préfixe p passe de 125 à 1254
16 -17	4 est retiré de G et retour à la ligne 9					
9	Nous sommes à la fin de la boucle, nous allons ligne 18					
18-19	Remise des sommets et arcs retirés dans G et retour ligne 3, $k = 3 \rightarrow$ ligne 19					
20	Renvoi de A sous la forme d'une liste : {1256, 1346, 12546}					

ANNEXE B

Résultats de l'analyse intégrative de l'implication du voisinage de gènes dans les processus biologiques d'*E. coli*

Table B.1 – Test du χ^2_1 sur le nombre de gènes essentiels et non essentiels entre k-$\mathcal{SIP}_{col}$ et $\mathcal{G}_{col}$.

	# gènes essentiels	# non-essentiels	χ^2 $\mathcal{G}_{col}$ v.s. k-$\mathcal{SIP}_{col}$
$\mathcal{G}_{col}$	135	644	-
1-$\mathcal{SIP}_{col}$	125	539	0,543
2-$\mathcal{SIP}_{col}$	130	577	0,283
3-$\mathcal{SIP}_{col}$	130	596	0,086
4-$\mathcal{SIP}_{col}$	131	605	0,058
5-$\mathcal{SIP}_{col}$	131	610	0,032
6-$\mathcal{SIP}_{col}$ à 10-$\mathcal{SIP}_{col}$	131	615	0,014

Table B.2 – Test statistique sur les fréquences d'apparition de gènes essentiels et non-essentiels qui interviennent dans les k-SIPs de $\mathcal{G}_{col}$. Le test de Wilcoxon permet d'évaluer si les gènes essentiels apparaissent de façon plus fréquente que les gènes non essentiels.

k-$\mathcal{SIP}_{col}$	Nombre de gènes		test de Wilcoxon		Interprétation statistique
	essentiels	non-essentiels	p-valeur	W_S	
1-$\mathcal{SIP}_{col}$	125	539	0,5845	34744,5	rejet H_0 impossible
2-$\mathcal{SIP}_{col}$	130	577	0,5618	38726,0	rejet H_0 impossible
3-$\mathcal{SIP}_{col}$	130	596	0,3755	40660,5	rejet H_0 impossible
4-$\mathcal{SIP}_{col}$	131	605	0,3718	41598,5	rejet H_0 impossible
5-$\mathcal{SIP}_{col}$	131	610	0,3656	41966,5	rejet H_0 impossible
6-$\mathcal{SIP}_{col}$	131	615	0,3502	42375,0	rejet H_0 impossible
7-$\mathcal{SIP}_{col}$	131	615	0,4014	42162,0	rejet H_0 impossible
8-$\mathcal{SIP}_{col}$	131	615	0,4165	42102,5	rejet H_0 impossible
9-$\mathcal{SIP}_{col}$	131	615	0,4903	41828,0	rejet H_0 impossible
10-$\mathcal{SIP}_{col}$	131	615	0,5165	41736,0	rejet H_0 impossible

Table B.3 – Les opérons d'*E. coli* qui correspondent exactement à au moins un 1-SIP chacun. Pour chaque opéron (première colonne), nous présentons seulement le 1-SIP (deuxième colonne) qui a le plus petit $\overline{w}_d$ (cinquième colonne) lui correspondant exactement. Les troisième et quatrième colonnes donnent respectivement le nombre de gènes et de réactions qui composent le 1-SIP. Dans la première colonne, le signe .+ signifie que l'ordre des gènes dans le 1-SIP suit le sens du brin principal, .- signifie que l'ordre des gènes suit le sens inverse du brin principal, . ? signifie que l'ordre des gènes ne correspond à aucun brin, et .+/- signifie que le 1-SIP est soit .+ soit .- mais qu'il existe d'autres 1-SIPs, avec un $\overline{w}_d$ plus grand, qui suivent le sens inverse.

Opéron	1-SIP	# gènes	# réact.	$\overline{w}_d$
fadIJ.+/-	(−b2341 :fadJ, R03276) → (−b2341 :fadJ, R03026) → (−b2341 :fadJ, R01975) → (−b2342 :fadI, R00238) → (−b2342 :fadI, R03778) → (−b2342 :fadI, R01177)	2	6	0.17
fadBA.+	(−b3846 :fadB, R04204) → (−b3846 :fadB, R04203) → (−b3845 :fadA, R00927) → (−b3845 :fadA, R03858) → (−b3845 :fadA, R04742) → (−b3845 :fadA, R01177)	2	6	0.17
hemCDXY.+	(−b3805 :hemC, R00084) → (−b3804 :hemD, R03165) → (−b3803 :hemX, R03194) → (−b3803 :hemX, R03947) → (−b3803 :hemX, R02864)	3	5	0.40
xylAB.+	(−b3565 :xylA, R01432) → (−b3564 :xylB, R01639)	2	2	0.50
uxuAB.+	(+b4322 :uxuA, R05606) → (+b4323 :uxuB, R02454)	2	2	0.50
kbl.tdh.-	(−b3616 :tdh, R01465) → (−b3617 :kbl, R00371)	2	2	0.50
betIBA.+	(−b0311 :betA, R01025) → (−b0312 :betB, R02565)	2	2	0.50
fucPIKUR.+	(+b2802 :fucI, R03163) → (+b2803 :fucK, R03241)	2	2	0.50
proBA.+	(+b0242 :proB, R00239) → (+b0243 :proA, R03313)	2	2	0.50
guaBA.+	(−b2508 :guaB, R08240) → (−b2507 :guaA, R08244)	2	2	0.50
garPLRK.rnpB.+	(−b3126 :garL, R02754) → (−b3125 :garR, R01747) → (−b3125 :garR, R01745) → (−b3124 :garK, R01514)	3	4	0.50
pdhR.aceEF.lpd.+	(+b0114 :aceE, R00014) → (+b0114 :aceE, R03270) → (+b0115 :aceF, R02569) → (+b0116 :lpd, R07618)	3	4	0.50
bglGFB.+	(−b3722 :bglF, R05132) → (−b3721 :bglB, R05133)	2	2	0.50
ascFB.+	(+b2715 :ascF, R04394) → (+b2716 :ascB, R05134)	2	2	0.50
fruBKA.+	(−b2169 :fruB, R03232) → (−b2168 :fruK, R02071)	2	2	0.50
mtlADR.+	(+b3599 :mtlA, R02704) → (+b3600 :mtlD, R02703)	2	2	0.50
speAB.+	(−b2938 :speA, R00566) → (−b2937 :speB, R01157)	2	2	0.50
ydiNB.aroD.+	(+b1692 :ydiB, R02413) → (+b1693 :aroD, R03084)	2	2	0.50
cynTSX.-	(+b0340 :cynS, R03546) → (+b0339 :cynT, R00132)	2	2	0.50
csiD.lhgO.gabDTP.-	(+b2662 :gabT, R01648) → (+b2661 :gabD, R00713)	2	2	0.50
edd.eda.+	(−b1851 :edd, R02036) → (−b1850 :eda, R05605)	2	2	0.50
ackA.pta.+	(+b2296 :ackA, R00315) → (+b2297 :pta, R00230)	2	2	0.50
nagBACD.+	(−b0678 :nagB, R00765) → (−b0677 :nagA, R02059)	2	2	0.50
otsBA.-	(−b1896 :otsA, R02737) → (−b1897 :otsB, R02778)	2	2	0.50
thrLABC.+	(+b0002 :thrA, R01773) → (+b0003 :thrB, R01771) → (+b0004 :thrC, R01466)	3	3	0.67
rhaBAD. ?	(−b3903 :rhaA, R02437) → (−b3904 :rhaB, R03014) → (−b3902 :rhaD, R02263) → (−b3902 :rhaD, R01785)	3	4	0.75

Table B.3 – suite de la page précédente

Opéron	1-SIP	# gènes	# réact.	$\overline{w}_d$
dgoRKADT.-	(−b4478 :dgoD, R03033) → (−b3693 :dgoK, R03387) → (−b4477 :dgoA, R01064)	3	3	1.0
bcsABZC.+	(−b3533 :bcsA, R02889) → (−b3531 :bcsZ, R02886)	2	2	1.0
glgCAP. ?	(−b3428 :glgP, R02111) → (−b3430 :glgC, R00948) → (−b3429 :glgA, R02421)	3	3	1.0
araBAD. ?	(−b0062 :araA, R01761) → (−b0063 :araB, R02439) → (−b0061 :araD, R05850)	3	3	1.0
entCEBAH.+	(+b0593 :entC, R01717) → (+b0595 :entB, R03037) → (+b0596 :entA, R01505)	3	3	1.0
gcvTHP.+	(−b2905 :gcvT, R01221) → (−b2903 :gcvP, R03425)	2	2	1.0
rfbBDACX. ?	(−b2039 :rfbA, R02328) → (−b2041 :rfbB, R06513) → (−b2038 :rfbC, R06514) → (−b2040 :rfbD, R02777)	4	4	1.75

Table B.4 – Couples et triplets d'opérons en correspondance exacte avec au moins un k-SIP chacun, pour k entre 1 et 10. Pour chaque groupe d'opérons, nous montrons les valeurs de k pour lesquelles un k-SIP correspond exactement au groupe, la plus petite valeur de $\overline{w}_d$ atteinte par un tel k-SIP parmi toutes les valeurs de k, la p-valeur du groupe d'opérons dans Gene Ontology (GO) et la meilleure p-valeur obtenue pour un opéron composant le groupe. La dernière colonne est un commentaire à propos du groupe.

Groupe d'opérons	k-SIP	Plus petit $\overline{w}_d$	*p-valeur* de GO	Meilleure *p-valeur** d'un opéron	Commentaire
Couples					
ackA-pta, *atoDAEB*	3 - 4	31.20 (3-4)	1.087×10^{-05}	2.075×10^{-06}	-
ackA-pta,*fadIJ*	1 - 10	5.75 (1)	6.926×10^{-03}	6.926×10^{-03}	-
ascFB, *bglGFB*	2	1.00 (2)	1.066×10^{-04}	1.835×10^{-01}	*ascF* et *bglF* sont homologues et *ascB* et *bglB* aussi
atoDAEB, *fadIJ*	3 - 10		3.871×10^{-08}	2.075×10^{-06}	Agissent dans la fonction d'oxydation/dégradation d'acide gras
csiD-lhgO-gabDTP, *frdABCD*	3 - 10		2.845×10^{-05}	9.187×10^{-06}	-
cysDNC, *cysJIH*	3 - 10		1.705×10^{-17}	1.384×10^{-08}	Über-operon, CysB comme activateur commun
fadBA, *fadIJ*	2 - 10		4.317×10^{-07}	1.079×10^{-04}	Fonction homologue, répresseurs communs (ArcA et FadR)
fadBA, *pdhR-aceEF-lpd*	1 - 10		1.790×10^{-04}	1.079×10^{-04}	Répresseur commun (ArcA) entre *fadBA*, *aceEF* et *lpd* units
glyQS, *kbl-tdh*	2	42.67 (2)	8.298×10^{-06}	1.383×10^{-06}	-
ilvIH, *ivbL-ilvBN*	4	0 (4)	1.934×10^{-11}	8.891×10^{-07}	*ilvB, ilvH, ilvI, ilvJ* sont des gènes homologues. *ilvN* n'est pas un gène métabolique, il n'est pas pris en compte.
ilvIH , *pdhR-aceEF-lpd*	4 - 10		1.989×10^{-04}	1.037×10^{-05}	-
kbl-tdh, *thrLABC*	1 - 2	125.02 (2)	2.153×10^{-10}	5.271×10^{-09}	-
leuLABCD, *pdhR-aceEF-lpd*	8 - 10	7.57 (8-10)	5.402×10^{-10}	1.758×10^{-11}	-
uxaCA, *uxuAB*	1	589.75 (1)	3.897×10^{-11}	8.891×10^{-06}	Ils ont un répresseur (ExuR) et un activateur (Crp) communs
Triplets					
ascFB, *bglGFB*, *chbBCARFG*	5	5.50 (5)	6.909×10^{-07}	5.344×10^{-03}	*chbA, chbB, chbC*, *ascF* et *bglF* homologues et *chbF*, *ascB* et *bglB* sont homologues également
glyQS, *kbl-tdh*, *thrLABC*	3-4	109.75 (4)	2.939×10^{-09}	5.271×10^{-09}	*kbl-tdh, thrLABC* fusionné avec *glyQS, kbl-tdh*.

* *fadIJ* et *frdABCD* n'ont pas de *p-valeurs*.

Table B.5 – Modules de KEGG de *E. coli* qui correspondent exactement à au moins un 1-SIP chacun. Pour chaque module, nous présentons seulement le 1-SIP avec le plus petit $\overline{w}_d$ (cinquième colonne) qui correspond exactement au module. La troisième et la quatrième colonnes donnent le nombre de gènes et de réactions qui interviennent dans le 1-SIP.

Module de KEGG	1-SIP	# gènes	# réact.	$\overline{w}_d$
M00017	(+b1761 :gdhA, R00243) → (+b1761 :gdhA, R00248)	1	2	0
M00029	(+b3772 :ilvA, R00996) → (+b3769 :ilvM, R04673) → (+b3774 :ilvC, R05069) → (+b3774 :ilvC, R05068) → (+b3771 :ilvD, R05070) → (+b3770 :ilvE, R02199)	5	6	2
M00030	(−b2913 :serA, R01513) → (+b0907 :serC, R04173) → (+b4388 :serB, R00582)	3	3	944
M00032	(−b3607 :cysE, R00586) → (−b2421 :cysM, R00897)	2	2	565
M00033	(−b2601 :aroF, R01826) → (−b3389 :aroB, R03083) → (+b1693 :aroD, R03084) → (+b1692 :ydiB, R02413) → (+b0388 :aroL, R02412) → (+b0908 :aroA, R03460) → (−b2329 :aroC, R01714)	7	7	789
M00035	(+b2021 :hisC, R00694) → (+b2599 :pheA, R01373) → (+b2599 :pheA, R01715)	2	3	187.67
M00037	(+b2019 :hisG, R01071) → (+b2026 :hisI, R04035) → (+b2026 :hisI, R04037) → (+b2024 :hisA, R04640) → (+b2023 :hisH, R04558) → (+b2022 :hisB, R03457) → (+b2021 :hisC, R03243) → (+b2022 :hisB, R03013) → (+b2020 :hisD, R03012) → (+b2020 :hisD, R01163)	7	10	1.5
M00041	(+b2818 :argA, R00259) → (+b3959 :argB, R02649) → (+b3958 :argC, R03443) → (−b3359 :argD, R02283) → (−b3957 :argE, R00669)	3	3	749
M00042	(+b3960 :argH, R01086) → (+b3172 :argG, R01954) → (−b4254 :argI, R01398)	3	3	597.67
M00051	(+b3940 :metL, R00480) → (−b3433 :asd, R02291)	2	2	245.5
M00055	(+b2942 :metK, R00177) → (−b1961 :dcm, R04858)	2	2	464
M00056	(+b2942 :metK, R00177) → (−b1961 :dcm, R04858)	2	2	464
M00059	(−b3846 :fadB, R04204) → (−b3846 :fadB, R04203) → (−b3845 :fadA, R00927)	2	3	0.33
M00063	(+b2521 :sseA, R03105) → (−b0928 :aspC, R00896)	2	2	782.5
M00097	(+b1236 :galU, R00289) → (−b0759 :galE, R00291)	2	2	237.5
M00098	(−b3729 :glmS, R00768) → (−b3176 :glmM, R02060) → (−b3730 :glmU, R05332)	3	3	367
M00099	(−b2048 :cpsG, R01818) → (−b2049 :cpsB, R00883)	2	2	0.5
M00115	(−b2053 :gmd, R00888) → (−b2052 :fcl, R05692)	2	2	0.5
M00117	(−b3092 :uxaC, R01983) → (−b1521 :uxaB, R02555) → (−b3091 :uxaA, R01540) → (+b4322 :uxuA, R05606) → (+b4323 :uxuB, R02454) → (−b3092 :uxaC, R01482)	5	6	893.67
M00119	(+b1215 :kdsA, R03254) → (+b3198 :kdsC, R03350) → (+b0918 :kdsB, R03351)	3	3	1316
M00120	(+b3619 :rfaD, R05176) → (+b4331 :kptA, R05644) → (−b2340 :sixA, R05647) → (+b4331 :kptA, R05646) → (+b0222 :lpcA, R05645)	5	5	946.4
M00160	(+b1093 :fabG, R04533) → (+b0180 :fabZ, R04428) → (−b1288 :fabI, R04429) → (+b1095 :fabF, R04952) → (+b1093 :fabG, R04953) → (+b0180 :fabZ, R04954) → (−b1288 :fabI, R04955) → (+b1095 :fabF, R04957) → (+b1093 :fabG, R04536) → (+b0180 :fabZ, R04537) → (−b1288 :fabI, R04958) → (+b1095 :fabF, R04960) → (+b1093 :fabG, R04534) → (−b0954 :fabA, R04535) → (−b1288 :fabI, R04961) → (+b1095 :fabF, R04963) → (+b1093 :fabG, R04964) → (+b0180 :fabZ, R04965) → (−b1288 :fabI, R04724) → (+b1095 :fabF, R04726) → (+b1093 :fabG, R04566) → (+b0180 :fabZ, R04568) → (−b1288 :fabI, R04966) → (+b1095 :fabF, R04968) → (+b1093 :fabG, R04543) → (+b0180 :fabZ, R04544) → (−b1288 :fabI, R04969)	5	27	498.70
M00181	(−b4041 :plsB, R00851) → (−b3018 :plsC, R02241)	2	2	486.5
M00182	(−b4041 :plsB, R00851) → (−b3018 :plsC, R02241)	2	2	486.5
M00192	(−b0420 :dxs, R05636) → (+b0173 :dxr, R05688) → (−b2747 :ispD, R05633) → (−b1208 :ispE, R05634) → (−b2746 :ispF, R05637) → (−b2515 :ispG, R05883) → (+b0029 :ispH, R05884) → (+b2889 :idi, R01123)	8	8	1057.63
M00210	(−b1912 :pgsA, R01801) → (+b1278 :pgpB, R02029)	2	2	313.5
M00243	(−b2688 :gshA, R00894) → (+b2947 :gshB, R00497)	2	2	120
M00245	(−b3974 :coaA, R03018) → (+b3639 :dfp, R04230) → (+b3639 :dfp, R03269) → (+b3634 :coaD, R03035) → (−b0103 :coaE, R00130)	4	5	227.4
M00248	(+b0776 :bioF, R03210) → (−b0774 :bioA, R03231) → (+b0778 :bioD, R03182) → (+b0775 :bioB, R01078)	4	4	2.25
M00255	(+b1236 :galU, R00289) → (−b2028 :ugd, R00286)	2	2	388
M00269	(+b0048 :folA, R00936) → (+b0048 :folA, R00939)	1	2	0
M00295	(−b2926 :pgk, R01512) → (+b1779 :gapA, R01063) → (+b1779 :gapA, R01061)	2	3	362.67
M00680	(−b1850 :eda, R05605) → (+b3526 :kdgK, R01541)	2	2	793

Table B.6 – Couples de modules de KEGG qui correspondent exactement à au moins un k-SIP, pour k compris entre 1 et 10. Pour chaque couple de modules de KEGG nous présentons les valeurs de k pour lesquelles il existe un k-SIP correspondant exactement au couple de modules, la plus petite valeur de $\overline{w}_d$ atteinte par de tels k-SIPs, la p-valeur obtenue dans Ecycoc, avec `GO::TermFinder`, par le couple de modules via ses gènes, et la meilleure p-valeur obtenue par l'un des modules qui composent le couple.

Couple de modules	k-SIPs	Plus petit $\overline{w}_d$	p-valeur* de GO	Meilleure p-valeur* obtenue par un module
Couples				
M00017, M00022	1-2	994.33 (1)	6.223×10^{-06}	1.976×10^{-06}
M00017, M00041	1-2	542.33 (1)	2.459×10^{-13}	1.677×10^{-12}
M00017, M00243	1	375.33 (1)	> 0.05	NA
M00023, M00051	2-10	767.9 (2)	2.090×10^{-15}	2.090×10^{-15}
M00025, M00051	2-10	300.33 (2)	7.469×10^{-11}	7.469×10^{-11}
M00033, M00034	2-3	585.77 (2)	1.235×10^{-36}	1.531×10^{-23}
M00033, M00035	1	705.22 (1)	2.844×10^{-31}	1.531×10^{-23}
M00035, M00036	1-10	563.0 (1)	2.846×10^{-08}	2.846×10^{-08}
M00041, M00042	1-2	540.0 (1)	6.913×10^{-24}	1.678×10^{-12}
M00055, M00056	1-10	464.0 (1)	5.112×10^{-04}	5.112×10^{-04}
M00063, M00243	1-2	879.75 (1)	2.005×10^{-02}	3.386×10^{-02}
M00097, M00255	1-10	417.0 (1)	1.171×10^{-02}	3.956×10^{-03}
M00099, M00115	1-10	1.5 (1-10)	2.908×10^{-06}	4.535×10^{-03}
M00181, M00182	1-10	486.5 (1-10)	1.659×10^{-05}	1.659×10^{-05}
M00192, M00295	1-4	1088.7 (1)	6.136×10^{-19}	7.453×10^{-21}
M00295, M00680	1-4	686.5 (1)	4.590×10^{-03}	7.705×10^{-04}

* pour les gènes associés.

Table B.7 – Meilleurs k-SIPs pour les couples de modules de KEGG. Pour chaque couple de modules de KEGG, le k-SIP avec le plus petit $\overline{w}_d$ correspondant exactement au module est présenté en détail. Chaque symbole • indique un chemin différent qui compose le k-SIP (ainsi le nombre de • donne la valeur de k). Les caractères en gras indiquent des sous-chemins avec des gènes successifs sur le génome.

Couple de modules	k-SIP avec le plus petit $\overline{w}$
M00017, M00022	• (−b0674 :asnB, R00578) → (+b1761 :gdhA, R00243) → (+b1761 :gdhA, R00248) → (+b3744 :asnA, R00483)
M00017, M00041	• (+b1761 :gdhA, R00243) → (+b1761 :gdhA, R00248) → (+b2818 :argA, R00259) → **(+b3959 :argB, R02649)** → **(+b3958 :argC, R03443)**→ (−b3359 :argD, R02283) → (−b3957 :argE, R00669)
M00017, M00243	• (+b1761 :gdhA, R00243) → (+b1761 :gdhA, R00248) → (−b2688 :gshA, R00894) → (+b2947 :gshB, R00497)
M00023, M00051	• (+b3940 :metL, R00480) → (−b3433 :asd, R02291) → (−b2478 :dapA, R02292) → (+b0031 :dapB, R04199) → (−b0166 :dapD, R04365) → (−b3359 :argD, R04475) → (+b2472 :dapE, R02734) → (+b3809 :dapF, R02735) → (−b2838 :lysA, R00451) • (+b3940 :metL, R00480) → (−b3433 :asd, R02291) → (−b2478 :dapA, R02292) → (+b0031 :dapB, R04199) → (+b0031 :dapB, R04198) → (−b0166 :dapD, R04365) → (−b3359 :argD, R04475) → (+b2472 :dapE, R02734) → (+b3809 :dapF, R02735) → (−b2838 :lysA, R00451)
M00025, M00051	• (+b3940 :metL, R00480) → (−b3433 :asd, R02291) → (−b4024 :lysC, R01773) → **(+b0003 :thrB, R01771)** → **(+b0004 :thrC, R01466)** • (+b3940 :metL, R00480) → (−b3433 :asd, R02291) → (−b4024 :lysC, R01773) → (−b4024 :lysC, R01775) → **(+b0003 :thrB, R01771)** → **(+b0004 :thrC, R01466)**
M00033, M00034	• (−b2601 :aroF, R01826) → (−b3389 :aroB, R03083) → **(+b1693 :aroD, R03084)** → **(+b1692 :ydiB, R02413)**→ (+b0388 :aroL, R02412) → (+b0908 :aroA, R03460) → (−b2329 :aroC, R01714) → **(−b1263 :trpD, R00986)** → **(−b1263 :trpD, R01073)** → **(−b1262 :trpC, R03509)** → **(−b1262 :trpC, R03508)** → **(−b1261 :trpB, R02722)** • (−b2601 :aroF, R01826) → (−b3389 :aroB, R03083) → **(+b1693 :aroD, R03084)**→ **(+b1692 :ydiB, R02413)**→ (+b0388 :aroL, R02412) → (+b0908 :aroA, R03460) → (−b2329 :aroC, R01714) → **(−b1264 :trpE, R00985)** → **(−b1263 :trpD, R01073)** → **(−b1262 :trpC, R03509)** → **(−b1262 :trpC, R03508)** → **(−b1261 :trpB, R02722)**
M00033, M00035	• (−b2601 :aroF, R01826) → (−b3389 :aroB, R03083) → **(+b1693 :aroD, R03084)** → **(+b1692 :ydiB, R02413)**→ (+b0388 :aroL, R02412) → (+b0908 :aroA, R03460) → (−b2329 :aroC, R01714) → (+b2599 :pheA, R01715) → (+b2599 :pheA, R01373) → (+b2021 :hisC, R00694)
M00035, M00036	• (+b2021 :hisC, R00734) → (+b2599 :pheA, R01728) → (+b2599 :pheA, R01715) → (+b2599 :pheA, R01373) → (+b2021 :hisC, R00694)
M00041, M00042	• (+b2818 :argA, R00259) → **(+b3959 :argB, R02649)** → **(+b3958 :argC, R03443)**→ (−b3359 :argD, R02283) → (−b3957 :argE, R00669) → (−b4254 :argI, R01398) → (+b3172 :argG, R01954) → (+b3960 :argH, R01086)
M00055, M00056	• (+b2942 :metK, R00177) → (−b1961 :dcm, R04858)
M00063, M00243	• (+b2521 :sseA, R03105) → (−b0928 :aspC, R00896) → (−b2688 :gshA, R00894) → (+b2947 :gshB, R00497)
M00097, M00255	• (−b0759 :galE, R00291) → (+b1236 :galU, R00289) → (−b2028 :ugd, R00286)
M00099, M00115	• **(−b2048 :cpsG, R01818)** → **(−b2049 :cpsB, R00883)** → **(−b2053 :gmd, R00888)** → **(−b2052 :fcl, R05692)**
M00181, M00182	• (−b4041 :plsB, R00851) → (−b3018 :plsC, R02241)
M00192, M00295	• (−b2926 :pgk, R01512) → (+b1779 :gapA, R01063) → (+b1779 :gapA, R01061) → (−b0420 :dxs, R05636) → (+b0173 :dxr, R05688) → (−b2747 :ispD, R05633) → (−b1208 :ispE, R05634) → (−b2746 :ispF, R05637) → (−b2515 :ispG, R05883) → (+b0029 :ispH, R05884) → (+b2889 :idi, R01123)
M00295, M00680	• (−b2926 :pgk, R01512) → (+b1779 :gapA, R01063) → (+b1779 :gapA, R01061) → (−b1850 :eda, R05605) → (+b3526 :kdgK, R01541)

ANNEXE C

Résultats de l'analyse intégrative de l'implication de la coexpression de gènes dans les processus biologiques d'*E. coli*

Table C.1 – Test statistique entre les fréquences de gènes essentiels et non-essentiels qui interviennent dans les k-SIPs de k-$\mathcal{SIP}_{coexp}$.

k-$\mathcal{SIP}_{coexp}$	Nombre de gènes		test de Wilcoxon		Interprétation statistique
	essentiels	non-essentiels	p-valeur	W_S	
1-$\mathcal{SIP}_{coexp}$	103	471	0,0215	20841,5	rejet H_0
2-$\mathcal{SIP}_{coexp}$	107	514	0,1294	24938,0	rejet H_0 impossible
3-$\mathcal{SIP}_{coexp}$	108	531	0,2182	26520,0	rejet H_0 impossible
4-$\mathcal{SIP}_{coexp}$	111	536	0,1224	26979,0	rejet H_0 impossible
5-$\mathcal{SIP}_{coexp}$	112	539	0,0868	27082,5	rejet H_0 impossible
6-$\mathcal{SIP}_{coexp}$	112	545	0,1255	27177,0	rejet H_0 impossible
7-$\mathcal{SIP}_{coexp}$	112	548	0,1393	27969,0	rejet H_0 impossible
8-$\mathcal{SIP}_{coexp}$	112	551	0,1539	28220,5	rejet H_0 impossible
9-$\mathcal{SIP}_{coexp}$	112	552	0,1361	28152,5	rejet H_0 impossible
10-$\mathcal{SIP}_{coexp}$	112	553	0,1377	28215,5	rejet H_0 impossible

Table C.2 – Les opérons d'*E. coli* qui correspondent exactement à au moins un 1-SIP chacun. Pour chaque opéron (première colonne), nous présentons seulement le 1-SIP (deuxième colonne) lui correspondant exactement qui a le plus petit $\overline{w}_d$ (cinquième colonne). Les troisième et quatrième colonnes donnent respectivement le nombre de gènes et de réactions qui composent le 1-SIP. Dans la première colonne, le signe .+ signifie que l'ordre des gènes dans le 1-SIP suit le sens du brin principal, .- signifie que l'ordre des gènes suit le sens inverse du brin principal, . ? signifie que l'ordre des gènes ne correspond à aucun brin, et .+/- signifie que le 1-SIP est soit .+ soit .- mais qu'il existe d'autres 1-SIPs, avec un $\overline{w}_d$ plus grand, qui suivent le sens inverse.

Opéron	1-SIP	# gènes	# réact.	$\overline{w}_d$
araBAD. ?	(-b0062 :araA, R01761) → (-b0063 :araB, R02439) → (-b0061 :araD, R05850),	3	3	0.280
bcsABZC.-	(-b3533 :bcsA, R02889) → (-b3531 :bcsZ, R02886)	2	2	0.716
betIBA.+	(-b0311 :betA, R01025) → (-b0312 :betB, R02565)	2	2	0.293
csiD-lhgO-gabDTP.+	(+b2662 :gabT, R01648) → (+b2661 :gabD, R00713)	2	2	0.065
edd-eda.-	(-b1851 :edd, R02036) → (-b1850 :eda, R05605) → (-b1850 :eda, R00470) → (-b1850 :eda, R00471)	2	4	0, 662
fucPIKUR.+	(+b2802 :fucI, R03163) → (+b2803 :fucK, R03241)	2	2	0.738
guaBA.-	(-b2508 :guaB, R08240) → (-b2507 :guaA, R08244)	2	2	0.018
hemCDXY.-	(-b3805 :hemC, R00084) → (-b3804 :hemD, R03165) → (-b3803 :hemX, R03194) → (-b3803 :hemX, R03947) → (-b3803 :hemX, R02864)	2	2	0.061
kbl-tdh.+	(-b3616 :tdh, R01465) → (-b3617 :kbl, R00371)	2	2	0.236
nagBACD.+/-	(-b0678 :nagB, R00765) → (-b0677 :nagA, R02059)	2	2	0.036
otsBA.+	(-b1896 :otsA, R02737) → (-b1897 :otsB, R02778)	2	2	0.015
pdhR-aceEF-lpd.+	(+b0114 :aceE, R00014) → (+b0114 :aceE, R03270) → (+b0115 :aceF, R02569) → (+b0116 :lpd, R07618)	3	4	0.022
proBA.+	(+b0242 :proB, R00239) → (+b0243 :proA, R03313)	2	2	0.111
rhaBAD. ?	(-b3903 :rhaA, R02437) → (-b3904 :rhaB, R03014) → (-b3902 :rhaD, R02263) → (-b3902 :rhaD, R01785) → (-b3904 :rhaB, R01902)	3	5	0.005
speAB.-	(-b2938 :speA, R00566) → (-b2937 :speB, R01157)	2	2	0.073
thrLABC.+	(+b0002 :thrA, R01775) → (+b0002 :thrA, R01773) → (+b0003 :thrB, R01771) → (+b0004 :thrC, R01466)	3	4	0.099
uxuAB.+/-	(+b4323 :uxuB, R02454) → (+b4322 :uxuA, R05606)	2	2	1.522
xylAB.+/-	(-b3565 :xylA, R01432) → (-b3564 :xylB, R01639)	2	2	0.790
yacC-speED.+	(-b0120 :speD, R00178) → (-b0121 :speE, R01920)	2	2	0.022
ydiNB-aroD.+/-	(+b1692 :ydiB, R02413) → (+b1693 :aroD, R03084)	2	2	0.246

Table C.3 – Couples d'opérons en correspondance exacte avec au moins un k-SIP chacun, pour k entre 1 et 10. Pour chaque groupe d'opérons, nous donnons les valeurs de k pour lesquelles un k-SIP correspond exactement au groupe, la plus petite valeur de $\overline{w}_d$ atteinte par un tel k-SIP parmi toutes les valeurs de k, la p-valeur du groupe d'opérons dans GO et la meilleure p-valeur obtenue pour un opéron composant le groupe. La dernière colonne est un commentaire à propos du groupe.

Groupe d'opérons	k-SIP	Plus petit $\overline{w}_d$	*p-valeur* de GO	Meilleur *p-valeur* d'un opéron	Commentaire
Couples					
uxuAB, *uxaCA*	1	1.787 (1)	4.931×10^{-11}	1.000×10^{-05}	Crp en activateur commun et ExuR en répresseur commun, appartiennent au même processus biologique (processus métabolique et catabolique du glucuronate)
ilvIH, *ivbL-ilvBN*	4	0 (4)	1.934×10^{-11}	8.891×10^{-07}	*ilvB, ilvH, ilvI, ilvJ* sont des gènes homologues. *ilvN* n'est pas un gène métabolique, il n'est pas pris en compte.

Table C.4 – Modules de KEGG pour *E. coli* qui correspondent exactement à au moins un 1-SIP chacun. Pour chaque module, nous présentons seulement le 1-SIP avec le plus petit $\overline{w}_d$ (cinquième colonne) qui correspond exactement à un module. Les troisième et quatrième colonnes donnent le nombres de gènes et de réactions qui interviennent dans le 1-SIP.

Module de KEGG	1-SIP	# gènes	# réact.	$\overline{w}_d$
M00017	(+b1761 :gdhA, R00243) → (+b1761 :gdhA, R00248)	1	2	0
M00022	(+b3744 :asnA,R00483) → (−b0674 :asnB,R00578)	2	2	0.666
M00030	(−b2913 :serA, R01513) → (+b0907 :serC, R04173) → (+b4388 :serB, R00582)	3	3	1.340
M00032	(−b3607 :cysE, R00586) → (−b2421 :cysM, R00897)	2	2	0.974
M00033	(+b1704 :aroH,R01826) → (−b3389 :aroB,R03083) → (+b1693 :aroD,R03084) → (+b1692 :ydiB,R02413) → (−b3390 :aroK,R02412) → (+b0908 :aroA,R03460) → (−b2329 :aroC,R01714)	7	7	0.562
M00035	(+b2599 :pheA,R01715)→ (+b2599 :pheA,R01373) → (+b2021 :hisC,R00694)	2	3	0.0116
M00037	(+b2019 :hisG, R01071) → (+b2026 :hisI, R04035) → (+b2026 :hisI, R04037) → (+b2024 :hisA, R04640) → (+b2023 :hisH, R04558) → (+b2022 :hisB, R03457) → (+b2021 :hisC, R03243) → (+b2022 :hisB, R03013) → (+b2020 :hisD, R03012) → (+b2020 :hisD, R01163)	7	10	0.131
M00041	(+b2818 :argA, R00259) → (+b3959 :argB, R02649) → (+b3958 :argC, R03443) → (−b3359 :argD, R02283) → (−b3957 :argE, R00669)	3	3	0.323
M00051	(−b4024 :lysC,R00480) → (−b3433 :asd, R02291)	2	2	0.110
M00059	(−b3846 :fadB, R04204) → (−b3846 :fadB, R04203) → (−b3845 :fadA, R00927)	2	3	0.0212
M00063	(+b2521 :sseA, R03105) → (−b0928 :aspC, R00896)	2	2	1.021
M00097	(+b1236 :galU, R00289) → (−b0759 :galE, R00291)	2	2	0.481
M00099	(−b2048 :cpsG, R01818) → (−b2049 :cpsB, R00883)	2	2	0.087
M00117	(−b3092 :uxaC, R01983) → (−b1521 :uxaB, R02555) → (−b3091 :uxaA, R01540) → (+b4322 :uxuA, R05606) → (+b4323 :uxuB, R02454) → (−b3092 :uxaC, R01482)	5	6	1.365
M00120	(+b3619 :rfaD, R05176) → (+b4331 :kptA, R05644) → (−b2340 :sixA, R05647) → (+b4331 :kptA, R05646) → (+b0222 :lpcA, R05645)	4	5	0.255
M00210	(−b1912 :pgsA, R01801) → (b0418 :pgpA, R02029)	2	2	0.181
M00243	(−b2688 :gshA, R00894) → (+b2947 :gshB, R00497)	2	2	0.378
M00295	(−b2926 :pgk, R01512) → (+b1779 :gapA, R01063) → (+b1779 :gapA, R01061)	2	3	0.508
M00298	(−b1479 :maeA,R00216)→(−b1479 :maeA,R00214)	1	2	0
M00680	(−b1850 :eda, R05605) → (+b3526 :kdgK, R01541)	2	2	0.812

Table C.5 – Couples et triplets de modules de KEGG qui correspondent exactement à au moins un k-SIP, pour k compris entre 1 et 10. Pour chaque couple et triplets de modules de KEGG nous présentons les valeurs de k pour lesquelles il existe un k-SIP correspondant exactement au module, la plus petite valeur de $\overline{w}_d$ atteinte par de tels k-SIPs, la p-valeur obtenue dans Gene Ontology par le couple ou triplet de modules via ses gènes, et la meilleure p-valeur obtenue par l'un des modules qui composent le groupe.

Modules	k-SIPs	Plus petit $\overline{w}_d$	*p-valeur** de GO	Meilleure *p-valeur** obtenue par un module
Couples				
M00017, M00041	1-2	0.493	2.459×10^{-13}	1.677×10^{-12}
M00023, M00051	1-8	0.588	2.090×10^{-15}	2.090×10^{-15}
M00025, M00051	1-10	0.223	7.469×10^{-11}	7.469×10^{-11}
M00030, M00295	1-2	1.114	3.223×10^{-08}	1.572×10^{-09}
M00033, M00035	1 et 7	0.462	2.844×10^{-31}	1.531×10^{-23}
M00035, M00036	1-10	0.035	2.846×10^{-08}	2.846×10^{-08}
M00055, M00056	1-10	0.109	5.112×10^{-04}	5.112×10^{-04}
M00097, M00255	1-10	0.442	1.171×10^{-02}	3.956×10^{-03}
M00181, M00182	1-10	1.278	1.659×10^{-05}	1.659×10^{-05}
Triplet				
M00033, M00035, M00036	9-10	0.522	1.107×10^{-24}	1.531×10^{-23}

* pour les gènes associés.

Analyse des systèmes bactériens

une approche *in silico* pour intégrer les connaissances du vivant

Philippe BORDRON

Résumé

L'émergence des expériences dites à haut débit permet l'acquisition rapide de données concernant un système biologique. Les biologistes disposent ainsi, aujourd'hui, d'un nombre important de données de natures hétérogènes qu'ils cherchent à structurer et analyser. Les méthodes dites intégratives proposent de répondre à cette demande, mais la création d'une méthode générale et satisfaisant les requêtes précises des biologistes constitue une tâche ardue.
Ce mémoire s'inscrit dans cette problématique. Nous y abordons diverses méthodes d'intégration des aspects *omiques* (métaboliques, génomiques, transcriptomiques...) d'un système bactérien et nous proposons la nôtre, nommée SIPPER, qui est une méthode générique et flexible. SIPPER permet de retrouver de l'information biologique cohérente entre les différents aspects étudiés grâce à la construction d'un modèle intégratif et l'utilisation d'une distance reposant sur des propriétés ou hypothèses biologiques choisies. Nous avons appliqué SIPPER deux fois sur les données métaboliques et génomiques d'*E. coli*. La première application teste l'hypothèse *les chaînes de réactions successives du réseau métabolique sont catalysées à l'aide d'enzymes produites par des gènes proches sur le génome*, et la seconde teste l'hypothèse *les chaînes de réactions successives sont catalysées par des gènes dont l'expression est similaire*. Nous avons découvert, par ces expériences, des mesures caractérisant certaines entités biologiques comme la *densité génomique* qui permet l'identification d'opérons métaboliques. L'apport de l'intégration de données supplémentaires aux approches n'utilisant traditionnellement qu'un seul type d'information a également été illustré au travers de la génomique comparative. Nous avons ainsi élaboré M&W-IISCS$_{\mathcal{M}}$, une méthode qui calcule des intervalles communs maximaux ayant un fort intérêt *omique*.

Mots-clés : biologie intégrative, informations *omiques*, plus courts chemins, opérons, modules métaboliques, génomique comparative, intervalles de gènes, intervalles communs

Abstract

Nowadays, the emergence of high throughput experiments allows a large number of biological data to be available to biologists, data that they need to structure and analyze. Integrative approaches provide a way to respond to this demand, but the creation of a method that is general and that satisfies the precise requests of biologists is a difficult task.
This is the problem of this thesis. We present various approaches that integrate many *omic* aspects (metabolic, genomic, transcriptomic...) of a bacterial system, and we also propose our own method, called SIPPER, that is both generic and flexible. SIPPER allows us to find consistent biological information between distinct *omic* aspects by constructing an integrated model and using a distance that is based on given biological properties and hypotheses. We apply SIPPER twice on metabolic and genomic data from *E. Coli*. The first application tests the hypothesis that *the chains of successive reactions in a metabolic network are catalyzed by enzymes that are products of neighbours genes in the genome*, and the second application tests the hypothesis that *the chains of successive reactions in a metabolic network are catalyzed by genes that have a similar expression*. These experiments allow us to identify measures that describe some biological entities such as the *genomic density* that allows us to identify metabolic operons. Integrating different kinds of data into traditional approaches that used only one kind of information is also illustrated in comparative genomics. Thus, we have elaborated M&W-IISCS$_{\mathcal{M}}$, a method that computes maximum common intervals that have an important *omic* interest.

Keywords: integrative biology, *omic* information, shortest paths, operons, metabolic modules, comparative genomics, gene intervals, commun intervals