Ecole doctorale MSTII
UFR IMA

Reconstruction *ab initio* de voies métaboliques – Formalisation et approches combinatoires

THÈSE

présentée et soutenue publiquement le 9 juillet 2004

pour l'obtention du

Doctorat de l'université Joseph Fourier – Grenoble 1

(spécialité Informatique : Systèmes et Communication)

par

Frédéric Boyer

Composition du jury

Rapporteurs :	Daniel Kahn
	Jacques Nicolas
Examinateurs :	Christine Collet
	François Fages
Directeurs de thèse :	Laurent Trilling
	Alain Viari

Institut National de Recherche en Informatique et en Automatique — Unité Rhône-Alpes
Laboratoire Logiciels, Systèmes, Réseaux — UMR 5526

Remerciements

Cher Lecteur,

je t'invite à t'associer à moi pour remercier ceux qui ont permis que tu lises ce travaille :

– Alain Viari mérites plus que tout autre ma gratitude. Les raisons pour lesquelles je tiens à le remercier seraient ici trop longues à expliquer. Il faut cependant que tu saches, cher Lecteur, que si tu peux lire aujourd'hui cette page de remerciements, c'est grâce à lui (ou plutôt "à cause de lui").

– Laurent Trilling m'a fait découvrir la bioinformatique pendant mon stage de DEA, de ceci, je lui serai longtemps redevable. Je le remercie également de tout l'intérêt qu'il a porté à mon travail pendant ces quatre années. Je lui tire également mon chapeau pour avoir su conserver tout son calme lors des discussions où j'ai laissé transparaître un caractère pour le moins singulier de ma personne (certains ont été jusqu'à dire en public que je suis têtu !).

– Anne Morgat et Eric Coissac avec qui j'ai partagé mon bureau et de longues discussions. Ils ont su répondre à mes nombreuses questions et je leur dois une bonne partie des connaissances "biologiques" que j'ai acquises durant ces quatre années.

– Les autres co-locataires des bureaux que j'ai successivement occupés, ainsi que mes "voisins de palier", ont également droit à une pensée chaleureuse. Ils ont supporté l'ambiance de travail que je leur ai imposée et ce pendant presque deux ans pour certains d'entre eux !

Je remercie également tous ceux qui m'ont accompagné ou que j'ai croisés durant ces quatre années au sein du laboratoire, tous ceux qui m'ont soutenu, encouragé et donné l'envie de mener à terme ce travail (j'espère que je n'ai oublié personne). Peut-être en fais-tu d'ailleurs partie, cher Lecteur ? Quoi qu'il en soit, tu es le dernier qui a droit à mes remerciements. Saches que l'intérêt que tu portes à ce travail est la meilleure récompense que je pouvais espérer et de ceci je te remercie.

Table des matières

Table des figures

Avant-propos

Dans la dernière décennie, d'importants progrès technologiques ont permis aux biologistes d'obtenir des informations globales sur le fonctionnement de la cellule.

Si un grand nombre de techniques utilisées aujourd'hui sont parfaitement maîtrisées depuis de nombreuses années (pour le cas du séquençage par exemple, la technique utilisée aujourd'hui se base sur celle proposée par Sanger en 1977), leur utilisation est passée de l'échelle d'un gène ou d'une protéine à celle de la cellule. Ainsi, dans les années 90, les premières séquences de chromosomes complets ont été publiées.

D'autres technologies permettent aujourd'hui d'avoir une vision globale de la cellule :

- l'utilisation des puces à ADN permet de comparer l'expression des gènes dans une cellule ou un tissu cellulaire sous différentes conditions
- la technique du double-hybride permet de connaître les interactions entre protéines à l'échelle de la cellule
- le couplage de l'électrophorèse 2D ou de la chromatographie et de la spectrométrie de masse permet d'analyser l'ensemble des protéines ou des métabolites présents dans la cellule à un instant donné

L'afflux de ces données, qui a engendré la série des 'omes' (transcriptome, protéome, interactome, métabolome), permet de poser de nouvelles questions qui n'ont plus pour objectif un gène ou une protéine mais qui concerne plutôt les réseaux biologiques qui les unissent. Ces nouvelles données créent de nouveaux besoins tant au niveau de leur gestion que de leur analyse et modélisation. Ces nouveaux champs d'investigation concernent les voies de signalisation, le contrôle de la transcription ou les réseaux métaboliques. Au delà de la connaissance de ces réseaux indépendamment les uns des autres, un objectif important est la compréhension de leur fonctionnement concerté.

Dans le cadre de cette thèse, nous nous intéresserons aux voies métaboliques et en particulier au problème de leur reconstruction. Ce sujet est loin d'avoir été totalement exploré et les voies métaboliques sont loin d'être définitivement établies. En effet, les diagrammes disponibles représentant des voies métaboliques reflètent un consensus admis par la communauté sans véritablement en donner une définition précise. Qui plus est,

la définition de ce qu'est une voie métabolique n'est pas toujours très claire, et dépend souvent du biologiste interrogé. Cette thèse a donc un double objectif : il s'agit d'une part de proposer une formalisation (algorithmiquement exploitable) de ce qu'est une voie métabolique et d'autre part de fournir des algorithmes permettant au biologiste de les inférer. On distinguera trois étapes principales dans ce travail :

- la première étape est l'appropriation des connaissances biologiques nécessaires à la compréhension du problème posé. Dans le cas particulier qui nous intéresse, cela signifie l'étude des différentes ressources disponibles sur le métabolisme et l'acquisition des connaissances biologiques nécessaires à leur compréhension. L'étude des caractéristiques des données disponibles et leur confrontation à des modèles m'a permis, dans un premier temps, de mieux en comprendre la nature.
- la seconde étape, qui constitue le véritable travail à l'interface entre l'informatique et la biologie, est de définir de façon formelle l'objet biologique manipulé. La difficulté consiste alors à formaliser les concepts biologiques le plus simplement possible mais sans simplification excessive afin que les définitions obtenues restent réalistes. L'étude des définitions déjà proposées pour des problématiques similaires m'a permis de restreindre les champs d'investigation. Dans cette thèse, je proposerai ainsi, deux définitions différentes pour traiter le problème de la reconstruction de voies métaboliques
- la troisième étape, qui concerne le domaine de l'informatique, consiste en l'élaboration d'algorithmes efficaces pour résoudre le problème ainsi posé, en l'étude de leur propriétés et en leur implémentation

Ce manuscrit est organisé en quatre parties :

La première introduit les notions biologiques utilisées pour formuler les hypothèses sur lesquelles seront basés les travaux présentés. Elle présente également les données disponibles concernant le métabolisme.

La seconde partie se focalise sur les travaux antérieurs ainsi que ceux sur lesquels nous nous sommes basés pour développer notre propre approche.

Les deux parties suivantes présentent les travaux originaux qui constituent le corps de ce travail. La première de ces deux parties concerne les deux définitions différentes de que nous donnons pour une voie métabolique et présente, pour chacun de ces deux cas, une méthode de reconstruction.

La seconde partie constitue une amorce à l'étude des relations entre les réseaux. Ici, nous nous sommes intéressés aux relations qui existent entre un réseau métabolique et l'organisation des gènes sur un génome bactérien.

Première partie

Le métabolisme

Chapitre 1

Rappels biologiques

Ce chapitre comporte des définitions succinctes des notions de biologie qu'il est nécessaire de connaître pour bien appréhender ce travail.

Il est composé de deux parties, la première définit ce qu'est le métabolisme et la seconde donne un aperçu de comment les activités métaboliques peuvent être régulées.

1.1 Métabolisme et enzymes

Le métabolisme est le processus global qui assure aux organismes vivants l'apport et l'utilisation de l'énergie dont ils ont besoin pour assurer leurs fonctions. Pour ce faire, ils couplent les réactions exergoniques issues de l'oxydation des nutriments aux processus endergoniques nécessaires au maintien en vie, tels que l'accomplissement de travail mécanique, le transport actif de molécules contre des gradients de concentration et la biosynthèse de molécules complexes.

La plupart des réactions chimiques du métabolisme nécessite la présence d'enzymes pour avoir lieu correctement dans les conditions, notamment de pH et de température, offertes par la cellule. Le premier des deux paragraphes suivants se focalise sur les enzymes et leur classification. Le deuxième paragraphe présente le concept de voies métaboliques et en donne deux exemples.

1.1.1 Les enzymes et leur classification

Une enzyme est une protéine présentant des propriétés de catalyse d'une réaction chimique du métabolisme. Les enzymes augmentent les vitesses des réactions qu'elles catalysent en baissant la quantité d'énergie nécessaire pour activer la réaction, c'est-à-dire l'énergie qu'il faut fournir pour former le ou les intermédiaires instables de la réaction (voir

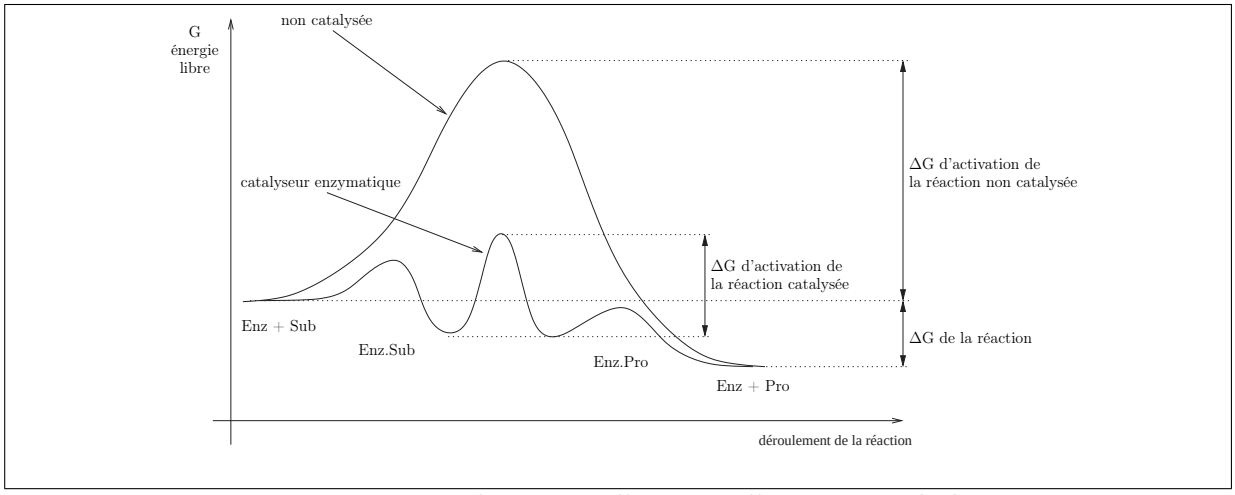


Fig. 1.1: Les enzymes diminuent l'énergie d'activation de la réaction

figure 1.1). Pour se faire, l'enzyme fixe les substrats à sa surface permettant de faciliter l'accès à l'état de transition.

Bien que soumises aux mêmes lois de la nature que les autres substances chimiques, les enzymes diffèrent des catalyseurs chimiques classiques sur plusieurs points importants :

- vitesse des réactions plus grandes : les vitesses des réactions catalysées par des enzymes sont multipliées par des facteurs compris entre 10^6 et 10^{12} par rapport aux réactions correspondantes sans catalyseurs, et sont au moins de plusieurs ordres de grandeur supérieurs aux réactions correspondantes catalysées par un catalyseur chimique,
- conditions de réactions plus douces : les réactions catalysées par des enzymes ont lieu dans des conditions relativement douces : température et pression atmosphérique physiologiques et pH proche de la neutralité, alors que les réactions sous la dépendance d'un catalyseur chimique nécessitent souvent des températures et des pressions élevées ainsi que des pH extrêmes,
- spécificité des réactions plus grandes : les enzymes ont des spécificités plus grandes vis-à-vis de leurs substrats (réactifs) et de leurs produits que les catalyseurs chimiques ; ainsi les réactions enzymatiques ne donnent que rarement des produits secondaires,
- possibilité de régulation : les activités catalytiques de nombreuses enzymes varient en réponse aux concentrations de substances autres que leurs substrats

La dénomination des enzymes est définie par des règles de la nomenclature internationale. Chaque enzyme est associée à un identifiant dans la classification enzymatique, cet identifiant, appelé "numéro EC", permet de décrire la réaction catalysée. La classification enzymatique est une classification hiérarchique à 4 niveaux. Chaque identifiant est donc formé de 4 nombres, où chaque nombre représente un des niveaux de la classification, le

premier niveau étant le plus général de la hiérarchie et le quatrième le plus spécialisé.

La classification regroupe au premier niveau six grandes classes d'enzymes qui sont :

Classification	Type de réaction catalysée
1. Oxydoréductases	Oxydoréduction
2. Transférases	Transfert de groupes fonctionnels
3. Hydrolases	Hydrolyse
4. Lyases	Elimination de groupes et formation de doubles liaisons
5. Isomèrases	Isomérisation
6. Ligases	Formation de liaisons couplées à l'hydrolyse de l'ATP (adénosine triphosphate)

Les deuxième et troisième niveaux représentent les sous-classes. Le quatrième niveau est un nombre arbitraire différent pour chaque enzyme partageant les trois premiers niveaux de la classification.

Exemple : l'enzyme nommée peptidyl-L-amino acide hydrolase a comme numéro EC 3.4.17.1 qui signifie que :

(3) l'enzyme est une hydrolase,

(4) cette enzyme agit sur des liaisons peptidiques (peptidases),

(17) cette enzyme est une métallocarboxypeptidase (cette enzyme possède un ion Zn^{2+} indispensable à son activité catalytique)

Cette classification a pour but de permettre une identification non ambiguë des enzymes et est utilisée systématiquement par la communauté scientifique.

1.1.2 Les voies métaboliques

L'ensemble des réactions et des composés chimiques présents dans les organismes définit un réseau métabolique universel présenté sur la figure 1.2. Cette carte présente les transformations possibles des métabolites par les différentes réactions dont ils sont les substrats ou les produits. Sur cette carte métabolique se dessinent des sous-réseaux plus ou moins indépendants du reste du réseau, le plus souvent sous la forme de chemins entre deux molécules.

Les voies métaboliques sont souvent définies comme des séries de réactions chimiques successives qui à partir d'un ensemble de substrats forment des produits spécifiques. Le plus souvent, ces voies sont associées à des fonctions biologiques précises (comme la biosynthèse d'un acide aminé, la dégradation des sucres). La définition des voies métaboliques reste floue et dépend des experts et des organismes considérés. La figure 1.3 montre schématiquement différents sous-réseaux du réseau métabolique d'un organisme, chacun de ces sous-réseaux représente une ou plusieurs voies métaboliques.

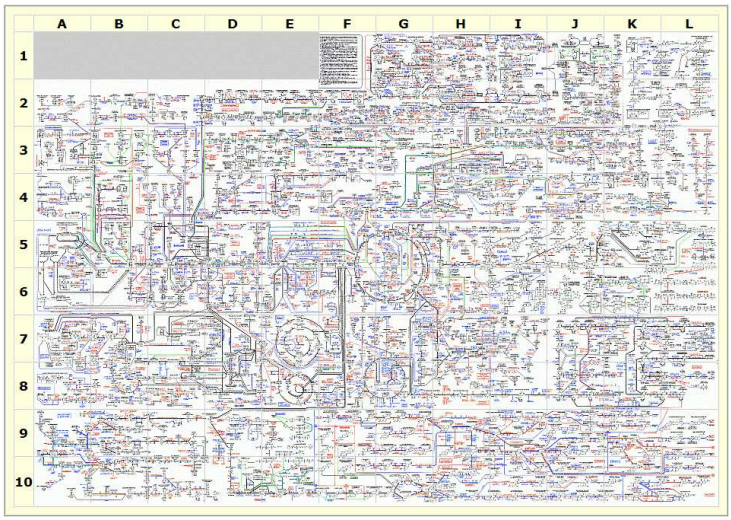


FIG. 1.2: La carte métabolique "Boehringer Mannheim Biochemical Pathways"

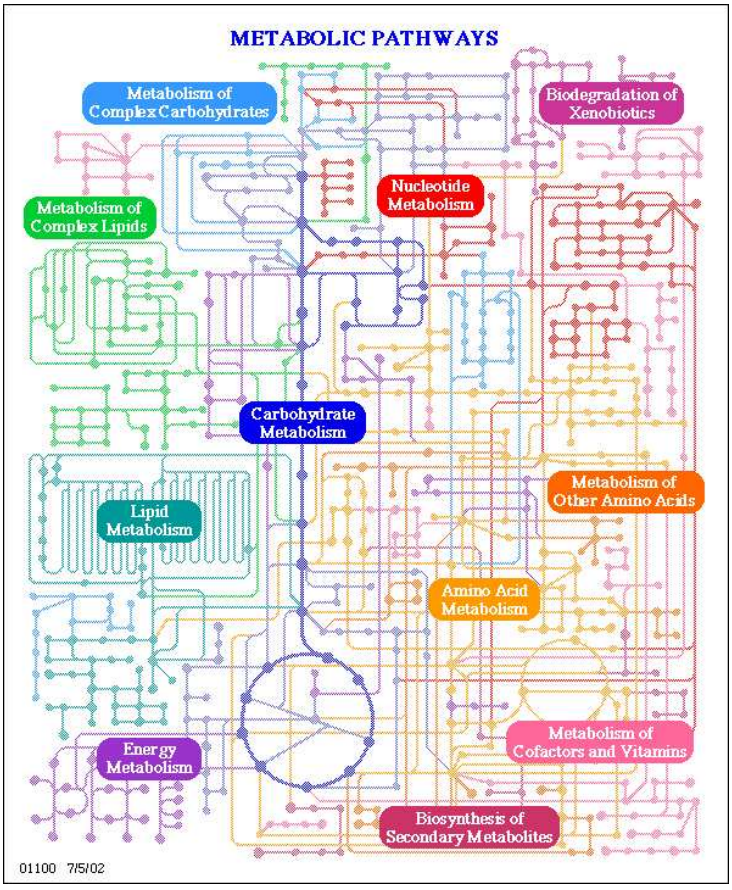


FIG. 1.3: Représentation schématique du réseau métabolique complet d'un organisme (extrait du site web de la banque KEGG `http ://www.genome.ad.jp/kegg/kegg2.html`)

On distingue deux grandes catégories de voies dans le métabolisme (voir figure 1.4) :

– les voies cataboliques sont les voies qui assurent la dégradation de molécules complexes en produits plus simples par des processus exergoniques. L'énergie libérée est utilisée pour la synthèse de l'ATP (adénosine triphosphate) à partir d'ADP (adénosine diphosphate) et de phosphate ou pour la réduction du $NADP^+$ (nicotinamide adénine dinucléotide phosphate) en NADPH. L'ATP et le NADPH sont les principales sources d'énergie libre pour les voies anaboliques.

– les voies anaboliques sont les voies qui à partir d'un ensemble restreint de composés simples (issus du catabolisme) synthétisent une multitude de produits variés nécessaires à la survie.

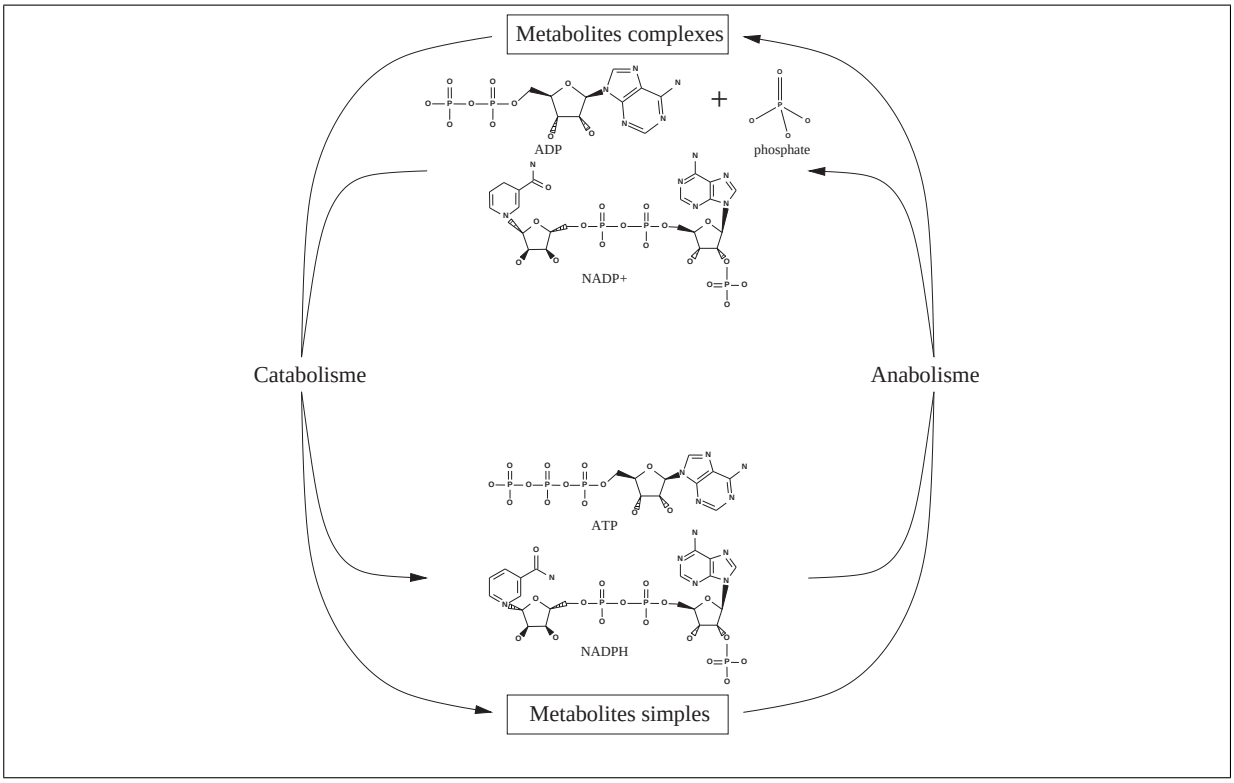


Fig. 1.4: L'ATP et le NADPH fournissent l'énergie libre nécessaire aux réactions de biosynthèses - Ils sont formés au cours de la dégradation de molécules complexes

1.1.2.1 Exemple d'une voie catabolique : la glycolyse

La glycolyse est une des principales voies métaboliques pour la fabrication d'énergie, c'est-à-dire la régénération des molécules d'ADP en ATP et de NAD^+ en NADH. Cette voie consiste en la transformation d'une molécule de glucose en deux molécules de pyruvate. La glycolyse peut être décomposée en deux parties. La première partie transforme la molécule de glucose en deux molécules de glycéraldéhyde 3-phosphate, une molécule à fort potentiel énergétique. Cette première partie est consommatrice d'énergie, est son

bilan est glucose + 2 ATP → 2 glycéraldéhyde 3-phosphate + 2 ADP. La seconde partie transforme les deux molécules de glycéraldéhyde 3-phosphate en deux molécules de pyruvate. Pour chaque molécule de glycéraldéhyde 3-phosphate, cette transformation permet la régénération de 2 molécules d'ADP en ATP et d'une molécule de NAD^+ en NADH. Le bilan de la deuxième partie est donc 2 glycéraldéhyde 3-phosphate + 4 ADP + 2 NAD^+ + 2 Pi + 2 H^+ → 2 pyruvate + 4 ATP + 2 NADH + 2 H^+ + 2 H_2O. Finalement, le bilan complet est donc glucose + 2 ADP + 2Pi + 2 NAD^+ → 2 pyruvate + 2 ATP + 2 NADH + 2 H^+ + 2 H_2O.

Cette voie est présente dans une grande partie des organismes.

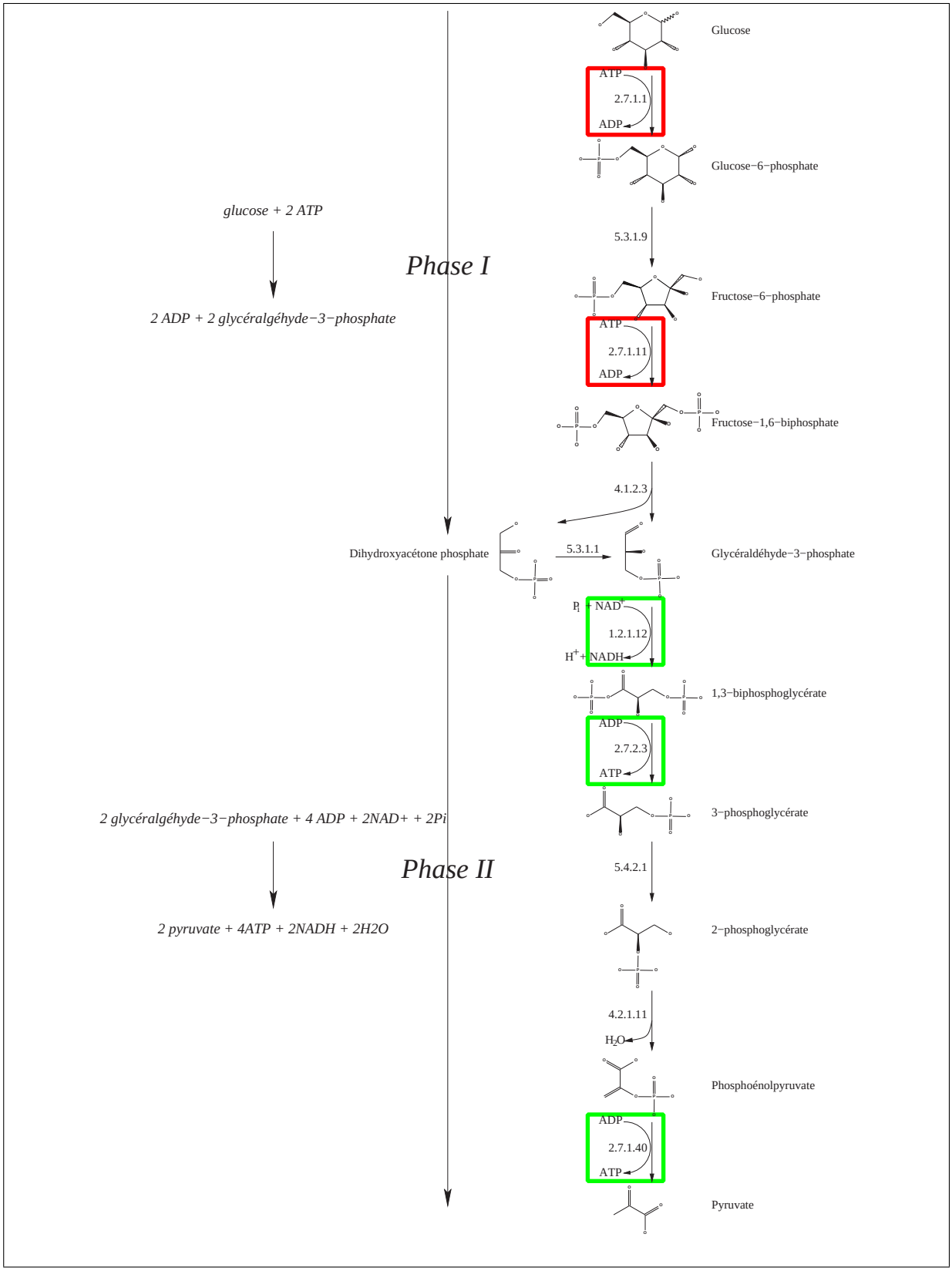


Fig. 1.5: Voie de dégradation du glucose en pyruvate : la glycolyse

1.1.2.2 Exemple d'une voie anabolique : biosynthèse du chorismate à partir de l'érythrose 4-phosphate et du PEP

Le chorismate est un composé aromatique qui sert de précurseur pour les trois acides aminés aromatiques phénylalanine, tyrosine et tryptophane. Ce composé est principalement fabriqué à partir de deux composés différents qui sont l'érythrose 4-phosphate et le PEP (phosphoenol pyruvate) (voir la figure 1.6).

Contrairement à la glycolyse qui est libératrice d'énergie, cette voie est largement déficitaire énergiquement, son bilan global est le suivant : 2 PEP + érythrose 4-phosphate + ATP + NADPH + H^+ → chorismate + ADP + 4 Pi + $NADP^+$+ H_2O. De plus, le PEP est un composé énergétiquement important car la dernière étape de la glycolyse transforme en effet une molécule de PEP en une molécule de pyruvate en régénérant un ADP en ATP. L'utilisation de PEP dans cette voie accentue donc encore le déficit énergétique de cette voie.

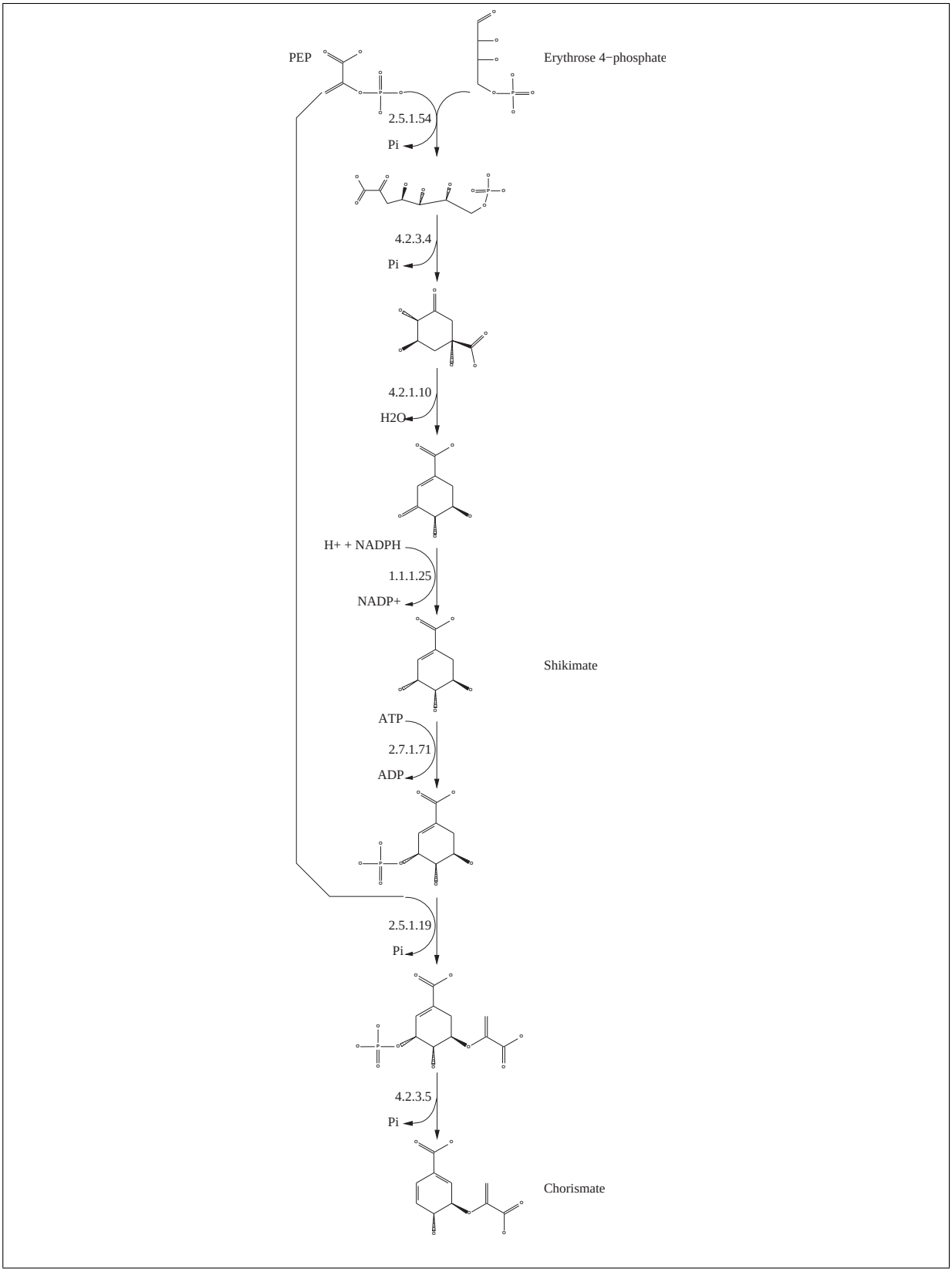


Fig. 1.6: La biosynthèse du chorismate à partir de l'érythrose 4-phosphate et du PEP

1.2 Régulation et métabolisme

Il n'est pas nécessaire ni souhaitable pour les organismes que toutes les fonctions métaboliques soient simultanément actives. Aussi, des mécanismes de régulation sont nécessaires au bon fonctionnement des organismes afin de synchroniser les fonctions métaboliques et de répondre à un changement de l'environnement.

Ces mécanismes de régulation peuvent intervenir à plusieurs niveaux. Chez les organismes bactériens, une grande partie de la régulation s'effectue au niveau de la transcription. L'organisation des génomes bactériens en opérons, que nous allons décrire dans la suite de ce chapitre, est une façon de mettre en œuvre efficacement la régulation transcriptionnelle. Cependant, comme le montrent quelques cas concrets comme les opérons lactose et arabinose, l'activation ou non des opérons peuvent être des phénomènes complexes.

1.2.1 Contrôle de la transcription

Des séquences de nucléotides particulières, sur la séquence d'ADN, sont reconnues par l'ARN polymérase qui s'y fixe pour commencer la transcription. Ces séquences particulières sont appelées "promoteurs". La séquence reconnue par l'ARN polymérase suit un modèle général. Chez les bactéries, ce modèle correspond à deux "boites" de séquence plus ou moins conservées situées à environ 35 et 10 bases du début de la transcription. En règle générale, l'affinité de la polymérase pour un promoteur dépend de la fidélité de celui-ci par rapport au modèle (voir figure 1.7). Cependant, l'affinité de l'ARN polymérase avec la séquence promotrice n'est pas le seul moyen de réguler la transcription. En effet, d'autres complexes moléculaires ont la capacité de reconnaître des séquences particulières sur l'ADN et de s'y fixer. La fixation de tels complexes peut entraîner une meilleure transcription du gène en amont si la fixation de ces complexes facilite l'amorçage de la transcription par la polymérase ou au contraire empêcher la transcription par la polymérase en empêchant l'accès du promoteur à la polymérase. Ces séquences particulières sur la molécule d'ADN sont appelées sites régulateurs et chaque site est la cible d'un complexe moléculaire particulier.

1.2.2 Organisation des génomes bactériens : les opérons

L'expression simultanée de plusieurs gènes garantit que leurs protéines vont être traduites en même temps. Dans le cas d'enzymes impliquées dans la même voie métabolique, le gain en efficacité est évident car cela garantit qu'aucune des étapes de la voie métabolique n'est manquante. L'organisation en opérons d'une fraction non négligeable des gènes sur les chromosomes bactériens reflète cette nécessité d'efficacité de la co-transcription de

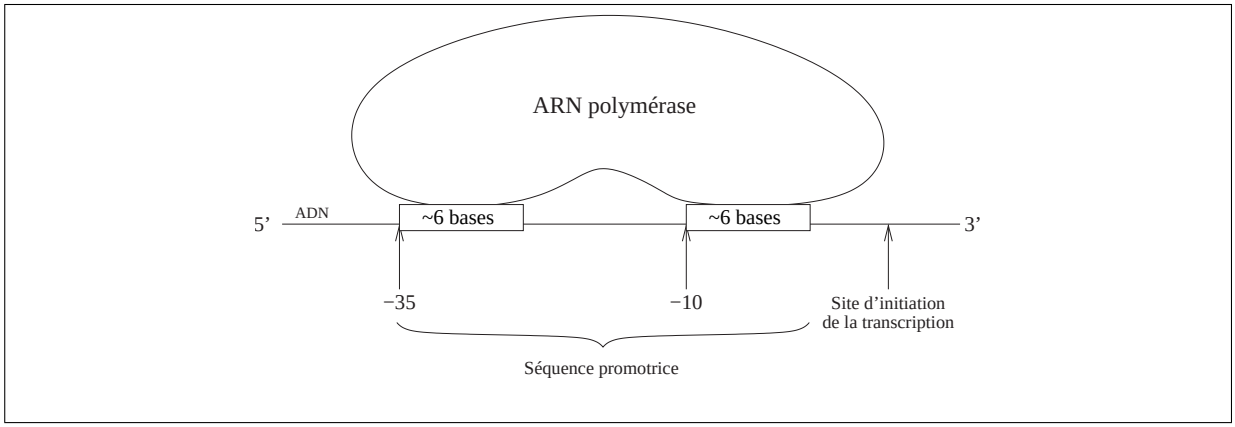


Fig. 1.7: Schéma simplifié d'un promoteur bactérien

certains gènes. Un opéron est un regroupement de plusieurs gènes co-orientés qui vont être transcrits simultanément donnant lieu à la synthèse d'un seul ARN messager (on parle d'ARN polycystronique) (voir la figure 1.8). Comme la traduction et la transcription sont étroitement liées chez les organismes bactériens, les protéines correspondantes seront donc, généralement, produites simultanément.

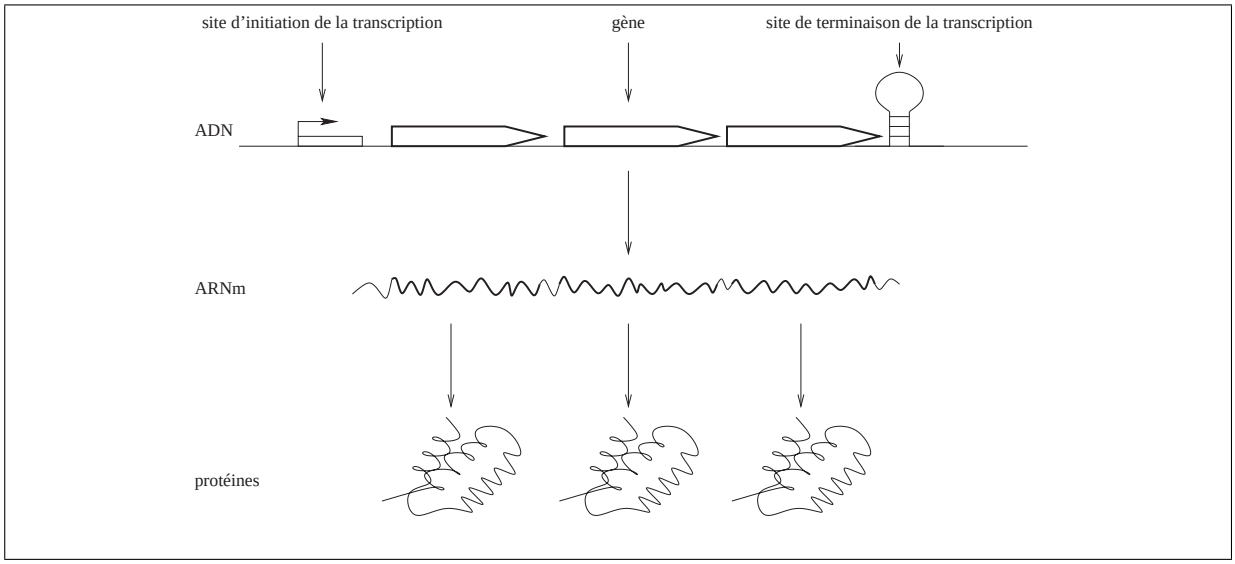


Fig. 1.8: La structure en opéron permet la transcription et la traduction simultanée de plusieurs gènes

1.2.3 Exemples d'opérons et de leur régulation

La régulation des deux opérons d'*Escherichia coli*, décrits dans les § 1.2.3.1 et 1.2.3.2, suivants montre bien la complexité de la régulation du métabolisme. Dans les deux cas, l'opéron contient les gènes codant pour des enzymes d'une même voie métabolique et est régulé entre autre par le produit d'un gène adjacent sur le chromosome.

Le glucose est le métabolite de choix d'*Escherichia coli*. La disponibilité de quantités suffisantes de glucose empêche l'expression de gènes codant pour des enzymes impliquées dans la transformation, en vue de la production d'énergie, de nombreux autres métabolites dont le lactose, le galactose et l'arabinose. Lorsque *Escherichia coli* est en présence de glucose, la concentration d'AMP cyclique (un métabolite) est fortement diminuée. Au contraire la diminution de glucose dans la cellule a pour effet d'augmenter la concentration d'AMP cyclique [Voet and Voet, 1998]. L'AMP cyclique est impliquée dans la régulation des deux opérons choisis pour illustrer leur régulation, il s'agit des opérons lactose et arabinose.

1.2.3.1 L'opéron lactose d'*Escherichia coli*

Les gènes de l'opéron lactose sont responsables de :

- l'import du lactose extracellulaire dans la cellule
- la transformation du lactose en glucose et galactose

Comme illustré sur la figure 1.9, en l'absence d'AMP cyclique (glucose en abondance) le produit du gène *lacI* vient se fixer à l'ADN et empêche la transcription de l'opéron lactose, la voie métabolique n'est donc pas active. Au contraire, lorsque l'AMP cyclique est présente, le produit du gène *lacI* se lie à l'AMP cyclique et devient incapable de se fixer à l'ADN. La transcription de l'opéron lactose peut alors avoir lieu permettant de faire pénétrer le lactose à une vitesse plus importante et de le transformer en glucose grâce aux produits des gènes *lacZ*, *lacY* et *lacA*.

L'opéron lactose a été le premier mécanisme de régulation de la synthèse des protéines mis en évidence [Jacob and Monod, 1961].

1.2.3.2 L'opéron arabinose d'*Escherichia coli*

L'opéron arabinose a un fonctionnement plus complexe. Comme pour l'opéron lactose, un des gènes en amont de l'opéron, le gène *araC*, code pour une protéine régulatrice de l'opéron. Cependant, contrairement au cas de l'opéron lactose, où *lacI* a un rôle uniquement de répresseur, *araC* présente à la fois un rôle de répresseur et d'activateur.

La régulation de cet opéron est illustrée sur la figure 1.10. Lorsque l'arabinose n'est pas présent dans le milieu, le gène *araC* est transcrit puis traduit en une protéine qui induit un changement de conformation de l'ADN empêchant alors l'accès du promoteur de l'opéron arabinose à l'ARN polymérase, bloquant ainsi la transcription. Au contraire, lorsque l'arabinose est présent ainsi que l'AMP-cyclique, la transcription de l'opéron arabinose est fortement induite en favorisant l'accès de la polymérase au promoteur de l'opéron.

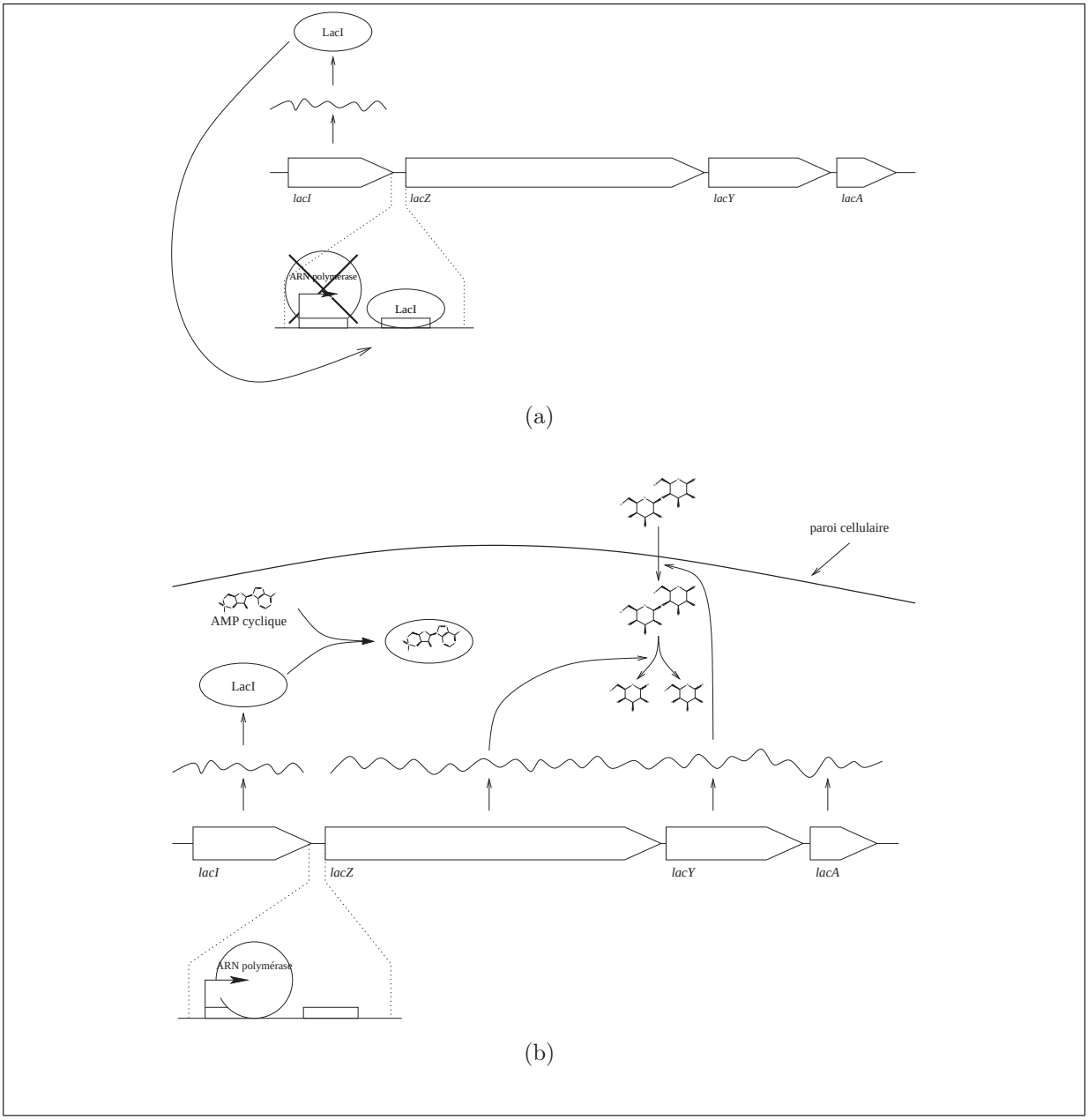


Fig. 1.9: L'opéron lactose d'*Escherichia coli* - (a) répression de l'opéron en absence d'AMP cyclique, (b) la présence d'AMP cyclique modifie l'affinité de la protéine LacI avec le promoteur de l'opéron lactose et permet la transcription de l'opéron (adapté de [Voet and Voet, 1998])

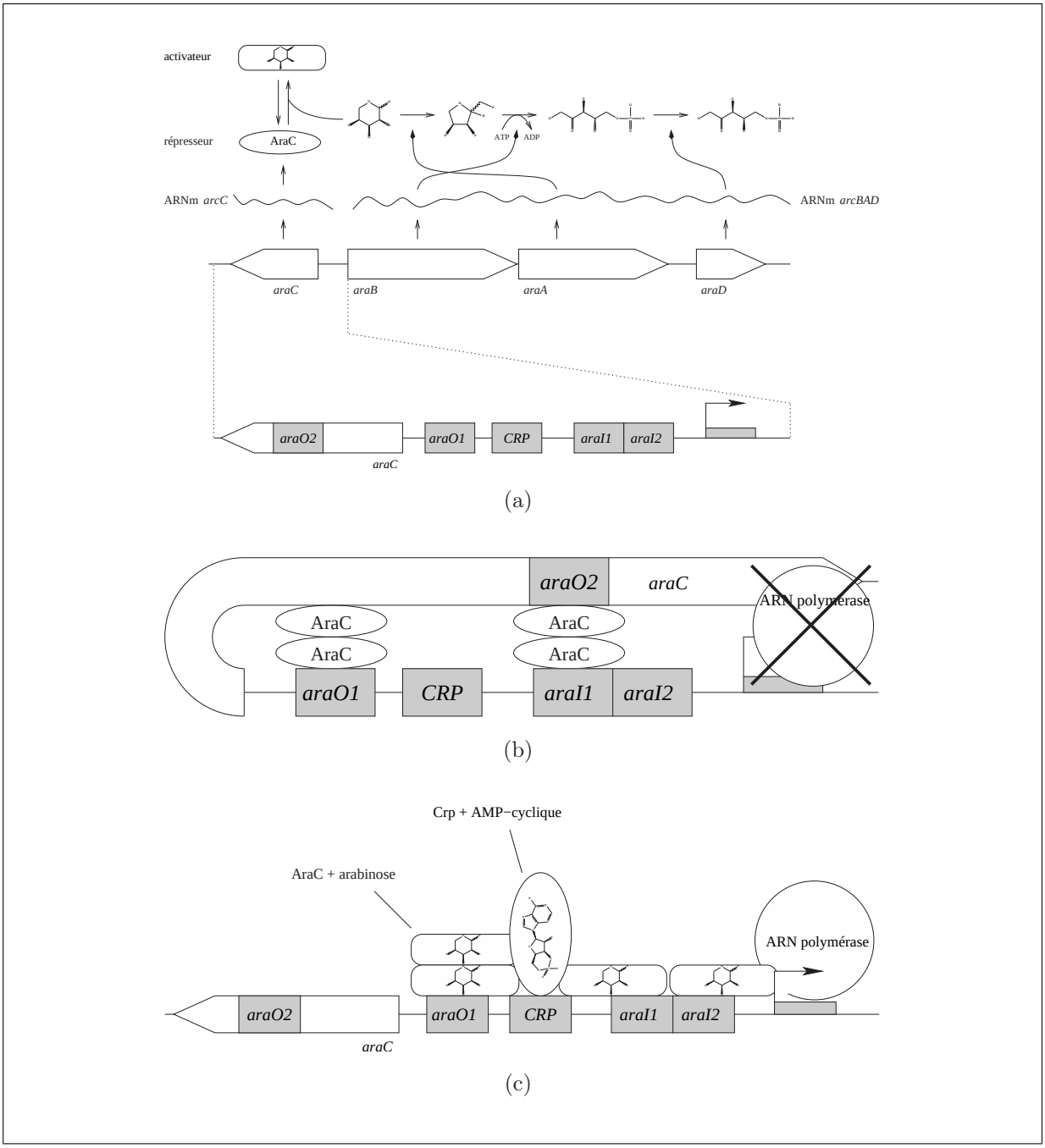


Fig. 1.10: L'opéron arabinose d'*Escherichia coli* - (a) le gène *araC* est le gène régulateur de l'opéron arabinose, les gènes *araA*, *araB* et *araD* catalysent des réactions successives intervenant dans la transformation de l'arabinose, (b) répression de l'opéron par l'absence d'arabinose, (c) activation de l'opéron par la présence d'arabinose et d'AMP cyclique (adapté de [Voet and Voet, 1998])

Chapitre 2

Banques et bases de connaissances dédiées au métabolisme

2.1 Les bases de connaissances EcoCyc/MetaCyc

La base de connaissances EcoCyc[Karp *et al.*, 2000; Karp and Riley, 1999] est dédiée à l'étude de la bactérie modèle *Escherichia coli*. Cette base décrit le génome et la machinerie biochimique d'*Escherichia coli*. Les données de la base proviennent de données de la littérature. Chaque voie métabolique connue fait l'objet d'une description détaillée ainsi que les enzymes impliquées (cofacteur, inhibiteur, assemblage moléculaire,...). Les composés impliqués dans les réactions sont également décrits individuellement, on a notamment accès à la structure 2D des composés.

La base MetaCyc [Karp *et al.*, 2002b] est dédiée à l'étude des voies métaboliques d'autres organismes (pas uniquement microbiens). Une autre base appelée BioCyc contient des prédictions de voies métaboliques pour des organismes dont la séquence du génome est complètement connue. Les prédictions se font sur le modèle de la "reconstruction par homologie" (voir § 4 - page 37). Ces deux bases partagent le modèle de données de la base EcoCyc.

Toutes ces bases possèdent la même interface de visualisation présentée sur la figure 2.1. La richesse des informations disponibles pour *Escherichia coli*, permet d'obtenir des diagrammes très détaillés qui, pour une voie métabolique donnée, montrent les réactions ainsi que les métabolites impliqués (en représentation 2D), les éventuels inhibiteurs des enzymes impliquées dans la voie métabolique et la localisation chromosomique des gènes codant pour les enzymes.

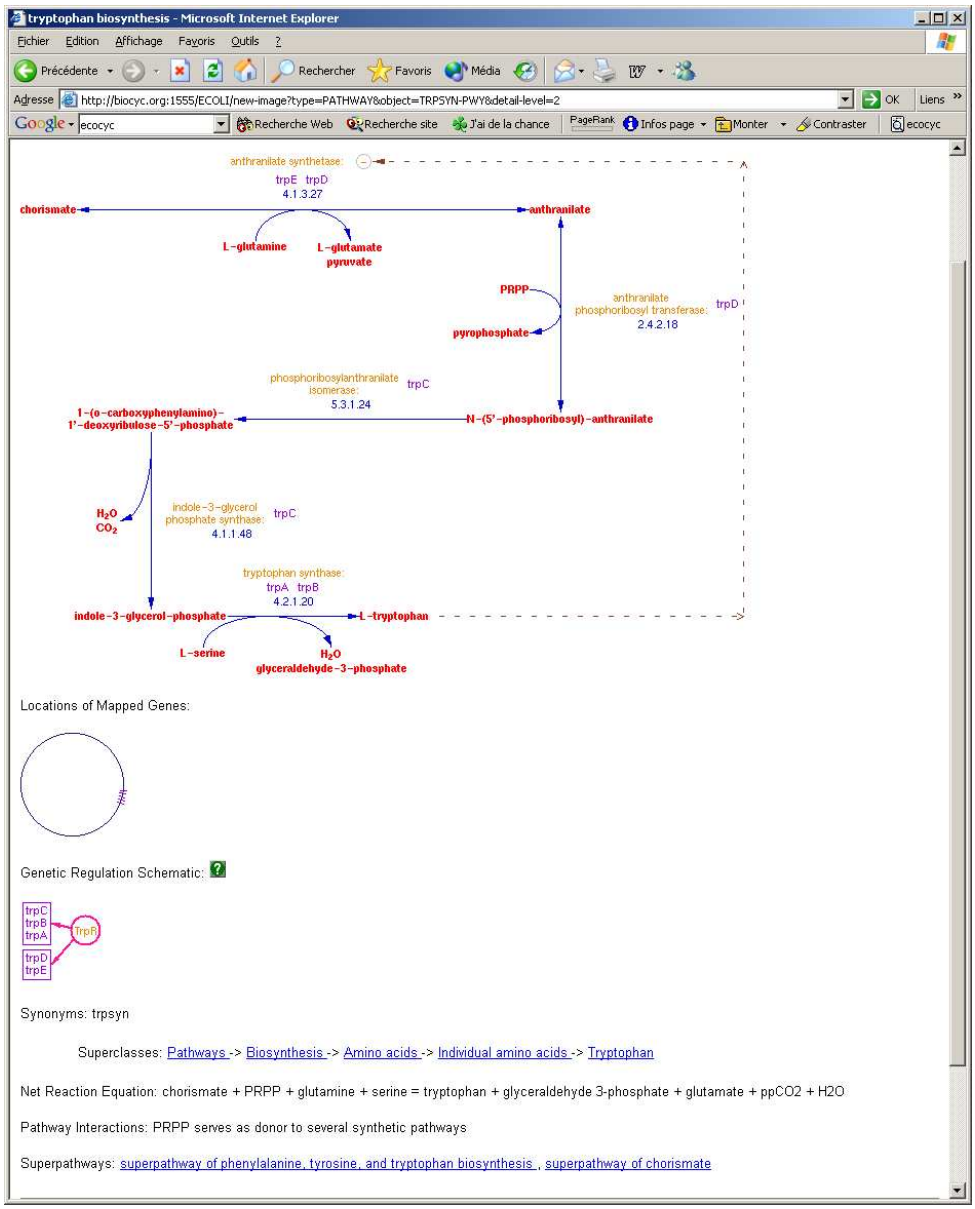


Fig. 2.1: Visualisation d'une voie métabolique dans EcoCyc

2.2 La banque Enzyme/Biochemical Pathways

La banque Enzyme [Bairoch, 2000] est un recueil de fiches écrit en langage semi-structuré (calqué sur le modèle de la banque Swiss-Prot **??**) concernant les réactions impliquées dans le métabolisme. Ces fiches sont annotées manuellement et sont associées à des références bibliographiques. Ces fiches contiennent des informations concernant la réaction et un ensemble d'identifiants de protéines catalysant cette réaction, répertoriées dans la banque Swiss-Prot.

Une version électronique de la carte métabolique "Boehringer Mannheim Biochemical Pathways" est également disponible sur le site Expasy (voir figure 1.2). Il est possible de visualiser des sous-parties de la carte. Sur cette carte, les composés chimiques sont représentés par leur formule développée, les enzymes sont représentées par leur nom usuel. Cette carte électronique est accessible par le web et un hyperlien sur chaque nom d'enzyme permet une connexion à la banque Enzyme.

2.3 La banque KEGG

La banque KEGG [Kanehisa *et al.*, 2002] (Kyoto Encyclopedia of Genes and Genomes) est le fruit d'un projet dédié à l'étude des processus biochimiques (métabolisme et régulation).

La banque KEGG est décomposée en trois parties. La première, nommée Ligand [Goto *et al.*, 2002], concerne les métabolites impliqués dans le métabolisme. Dans cette banque, chaque métabolite fait l'objet d'une description individuelle avec en particulier sa structure 2D. La deuxième partie concerne les réactions et les voies métaboliques. Dans KEGG, une voie métabolique est décrite par un ensemble de réactions impliquées dans la voie, sans distinction d'espèces. Ainsi, les voies métaboliques de KEGG ne sont pas des représentations consensuelles des voies métaboliques mais l'union des réactions impliquées dans un processus métabolique pour tous les organismes qui le prennent en charge (voir figure 2.2). La troisième partie concerne les gènes et les génomes des organismes avec en particulier l'assignation automatique aux gènes de fonctions enzymatiques. Cette partie contient des informations pour l'ensemble des génomes complètement séquencés.

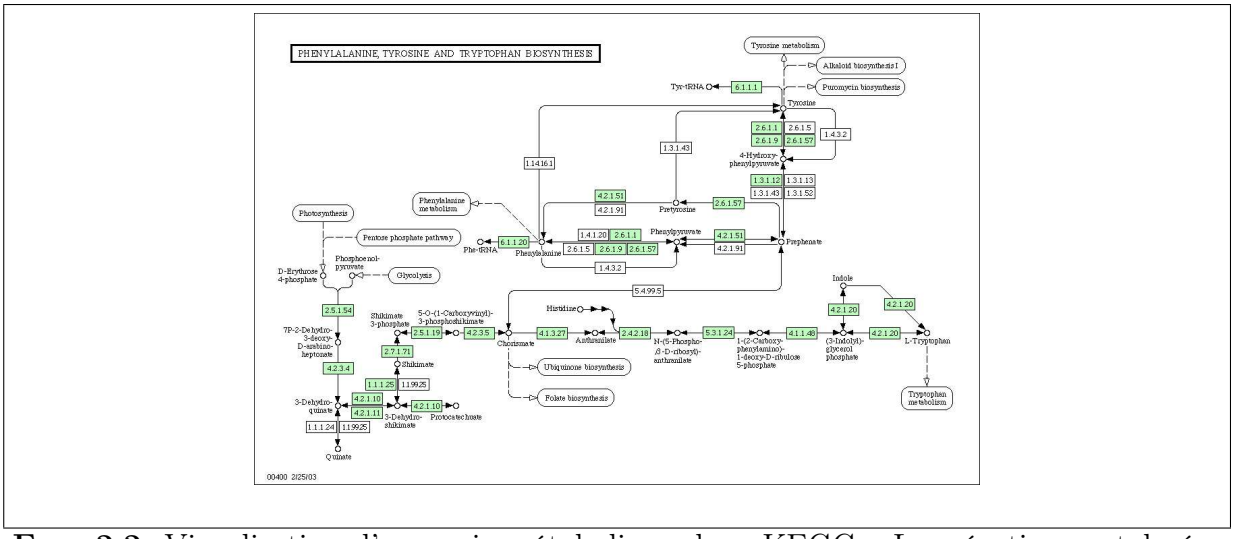


Fig. 2.2: Visualisation d'une voie métabolique dans KEGG - Les réactions catalysées par une enzyme présente chez *Escherichia coli* sont coloriées en vert

KEGG fournit sous forme de fichiers à plat, les données sur les voies métaboliques, les données sur les composés biochimiques et les réactions ainsi que sur les organismes et leurs gènes. Jusqu'à peu, KEGG était le seul projet mettant toutes ces données (voies métaboliques, réactions et composés, enzymes et génomes) à disposition de la communauté scientifique. Il faut noter que la qualité des données contenues dans KEGG laisse parfois à désirer. En effet, il y a un bon nombre de réactions incomplètes et certaines des structures de composés erronées. De plus, l'information est disséminée dans plusieurs fichiers et la cohérence n'est pas toujours assurée entre le contenu des différents fichiers.

Les contenus de ces trois bases, en ce qui concerne les réactions et leurs liens avec les numéros EC, sont à peu près identiques. La figure 2.3 montre la part des numéros EC présents dans les trois bases.

Bien qu'étant la banque contenant le plus de numéros EC et donc *a priori* le plus de réactions différentes, la banque Enzyme ne contient aucune section relative aux composés impliqués dans les réactions contenues dans la banque. De plus, une bonne part des champs décrivant les équations réactionnelles est écrite en langage naturel. Ces deux observations rendent l'extraction des réactions à partir de la banque Enzyme très difficile en l'absence de traitement manuel, ce qui n'est pas le cas pour les bases EcoCyc/MetaCyc et la banque KEGG.

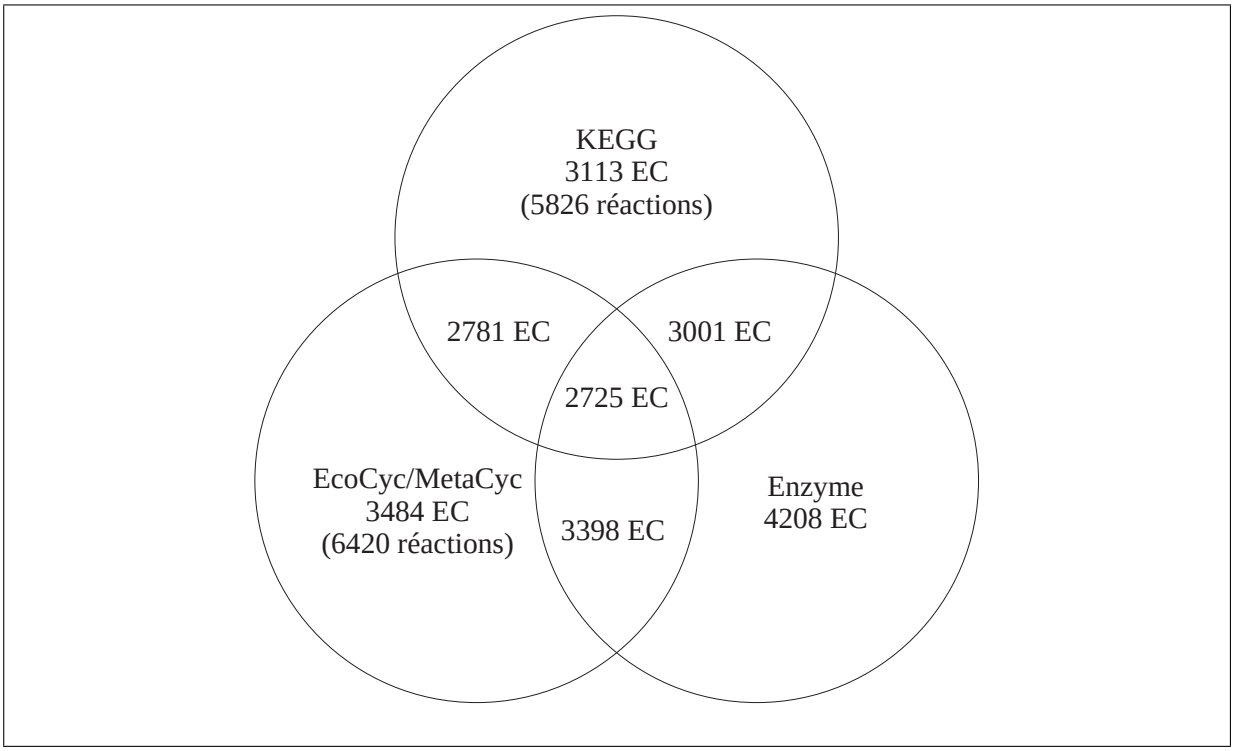


Fig. 2.3: Intersection des numéros EC contenus dans les bases EcoCyc/MetaCyc (février 2003), Enzyme (octobre 2003) et KEGG (novembre 2003)

2.4 Autres banques dédiées au métabolisme

Parmi les autres banques dédiées au métabolisme citons :

- Brenda [Schomburg *et al.*, 2002] : c'est une des banques de référence sur les enzymes et leur fonction dans le métabolisme. En particulier, il est possible d'accéder aux types des réactions (addition, élimination, substitution,...), ainsi qu'aux paramètres cinétiques des réactions. La liste des subtrats/produits potentiels est également beaucoup plus complète que celle des banques KEGG et Enzyme.
- WIT/EMP [Overbeek *et al.*, 2000] : cette base est accessible par un site web qui offre un ensemble d'outils dédiés à l'annotation de génomes totalement ou partiellement séquencés : reconstruction de voies métaboliques, résultats de génomique comparative... Il est également possible de travailler avec une séquence "privée" en débutant l'analyse par un criblage de la base EMP dédiée aux enzymes.
- UM-BBD [Ellis *et al.*, 2001] : l'objectif de cette base est de fournir des informations sur les réactions catalysées par des enzymes microbiennes d'intérêt pour le domaine des biotechnologies. Cette base couvre seulement une partie des réactions biochimiques du métabolisme intermédiaire.

Une description détaillée de toutes les bases de données dédiées au métabolisme, leur mode d'accès et de consultation ainsi que des exemples d'incohérences que l'on y trouve sont décrites dans [Wittig and De Beuckelaer, 2001]).

Deuxième partie

Etat de l'art

Cette partie a pour but de décrire successivement deux problèmes liés aux réseaux métaboliques : le problème de la reconstruction des réseaux métaboliques et le problème de leur caractérisation.

Le problème de la reconstruction consiste, pour un organisme donné, à reconstruire l'ensemble de ses voies métaboliques. Il y a encore quelques années, ce type de tâche était entrepris entièrement manuellement (voir par exemple [Bono *et al.*, 1998; Selkov *et al.*, 2000; 1997]). Avec le développement de ressources comme KEGG [Kanehisa *et al.*, 2002] ou WIT [Overbeek *et al.*, 2000], il est devenu possible de s'aider d'outils dans cette tâche de reconstruction manuelle. Néanmoins, avec l'arrivée massive de génomes complètement séquencés, le besoin d'outils permettant l'analyse et la reconstruction des voies métaboliques de manière automatique ou semi-automatique apparaît de plus en plus important.

Comme de plus en plus de réseaux métaboliques de grande taille sont disponibles, il devient possible d'étudier les caractères généraux des réseaux métaboliques et notamment leur topologie. Pour certains organismes comme *Escherichia coli*, le réseau métabolique est presque entièrement disponible. Cela permet de traiter des questions plus précises concernant en particulier les liens entre les voies métaboliques et l'organisation des gènes sur le génome.

Le problème de la reconstruction est lié à celui de la caractérisation, car la bonne connaissance des réseaux métaboliques permet d'expliquer les échecs de certaines méthodes de reconstruction et permet également de guider la conception de nouvelles méthodes.

Cette partie est séparée en 3 chapitres consacrés respectivement aux modèles utilisés pour manipuler les réseaux métaboliques, au problème de la reconstruction de voies métaboliques et à la caractérisation des réseaux métaboliques.

Chapitre 3

Modèles pour les graphes métaboliques

3.1 Représentation d'un ensemble de réactions par un graphe

La représentation sous forme de graphes d'un ensemble de réactions peut être très différente suivant les travaux [van Helden *et al.*, 2002].

La figure 3.1 montre différentes constructions associées au même ensemble de réactions. La représentation visuelle habituelle est montrée en (a).

Le graphe $\mathcal{G}_M$ (b) a autant de nœuds qu'il y a de métabolites dans le réseau original. Deux nœuds sont reliés s'ils sont l'un substrat et l'autre produit de la même réaction. Chaque arête est étiquetée par la réaction.

A chaque nœud du graphe $\mathcal{G}_R$ (c) est associée une réaction. Deux nœuds sont reliés si les deux réactions correspondantes partagent un substrat et un produit en commun.

Le graphe $\mathcal{G}_{RM}$ (d) est un graphe bipartite. Il a deux types de nœuds, un pour les métabolites, l'autre pour les réactions. Il n'y a d'arêtes qu'entre nœuds de types différents. Un nœud métabolite est relié par une arête aux nœuds réactions seulement si le métabolite est substrat ou produit de ces réactions. Un nœud réaction est relié aux nœuds métabolites qui sont ses substrats et produits.

Dans ces trois cas, il est possible d'orienter les arêtes. Dans ce cas, l'orientation doit bien tenir compte de la réversibilité des réactions.

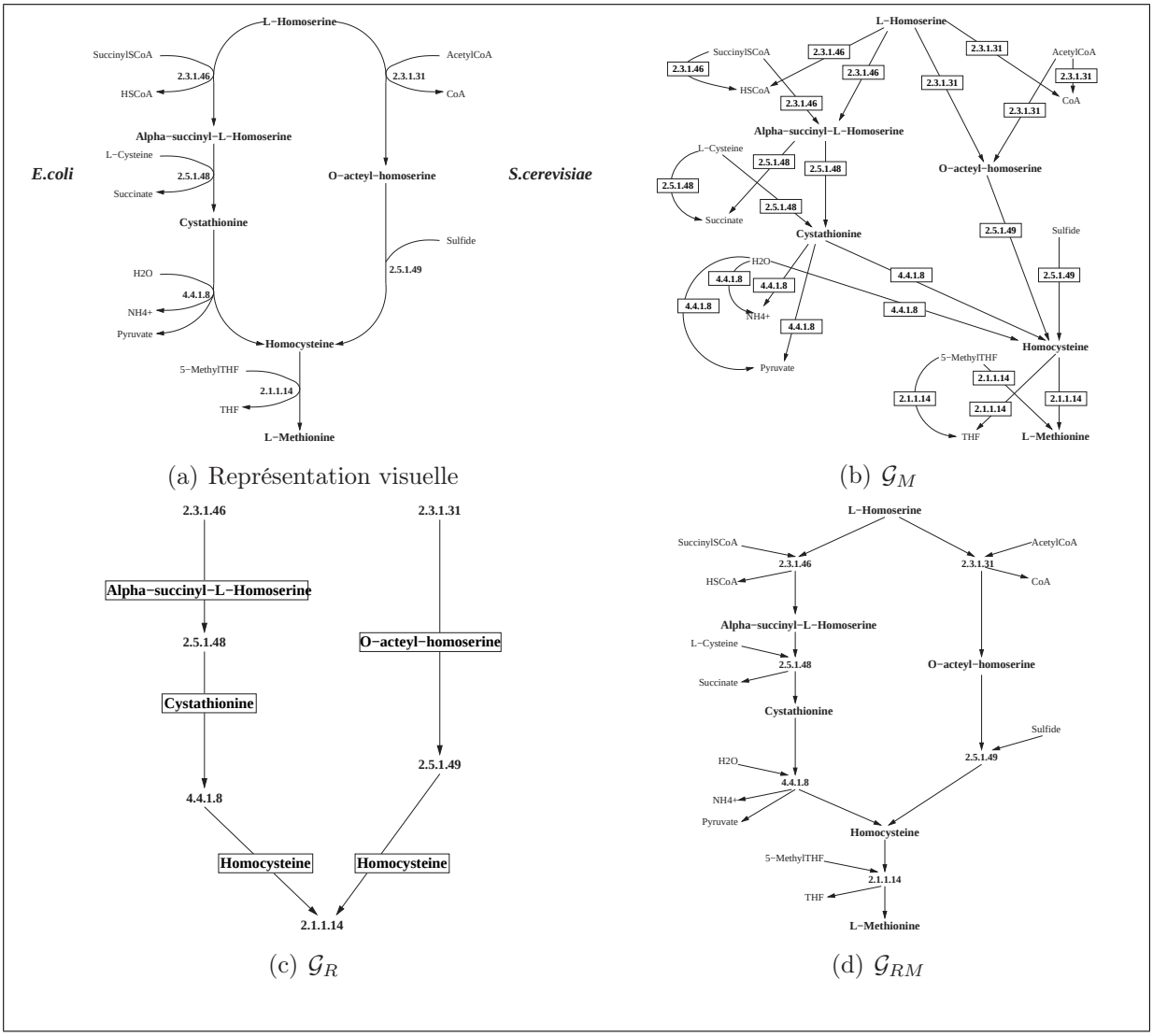


Fig. 3.1: Différents graphes construits à partir du même ensemble de réactions (adapté de [van Helden *et al.*, 2002]) - (a) représentation classique, (b) dans le graphe des composés, il y a un nœud par composé et tous les couples de composés $(substrat, produit)$ de la même réaction sont reliés par une arête, (c) dans le graphe des réactions, il y a un nœud par réaction et deux réactions sont reliées si elles ont un composé en commun, (d) graphe bipartite où chaque composé et réaction est associé à un nœud

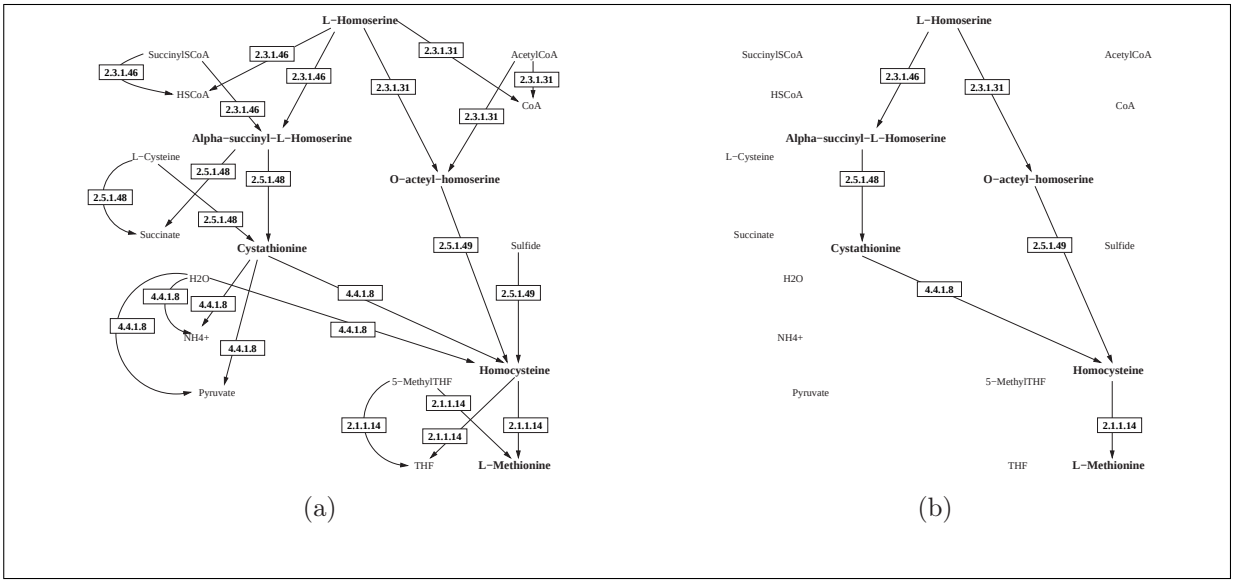


Fig. 3.2: Construction d'un graphe métabolique en tenant compte des composés primaires et secondaires - A partir du graphe des composés complet (a), il est possible de ne conserver que les relations entre les composés primaires des réactions pour donner un graphe des composés dont les chemins sont plus significatifs d'un point de vue biologique (b)

La définition systématique des liens entre les différents nœuds dans un graphe métabolique conduit le plus souvent à un graphe dont les chemins ne sont pas intéressants d'un point de vue biologique car ne reflétant aucunement la réalité de la transformation des composés. En effet, il faut tenir compte des composés qui interviennent dans de très nombreuses réactions comme les coenzymes, et les petits composés comme l'eau. La construction d'un graphe du type du graphe (b) de la figure 3.1 ne tient pas compte de ces composés. Il en ressort que deux nœuds représentant des composés différents ont une grande chance d'être à une faible distance car les composés fortement maillés (les coenzymes et les petits composés) font office de raccourci dans ce graphe. Communément, dans les représentations graphiques des voies métaboliques (comme le (a) de la figure 3.1) les composés dits *primaires* sont sur le chemin principal tandis que les composés dits *secondaires* sont mis visuellement en retrait. Il faut donc faire la distinction entre ces deux types de composés pour la construction du graphe. Il est possible de construire des sous-graphes où toutes les arêtes ne sont pas présentes. Par exemple, la figure 3.2 montre le graphe (b) de la figure 3.1 et le sous-graphe obtenu où seuls les composés primaires sont reliés entre eux. Etablir cette distinction entre composés primaires et secondaires demande soit un traitement automatique soit un traitement par un expert.

Dans le cas du traitement automatique, une liste de composés habituellement non primaires, comme l'eau, le CO_2, l'ammoniac et une liste de coenzymes comme l'ATP,

l'ADP, CoA ... est utilisée. Dans [Ma and An-Ping, 2003], l'ensemble des réactions de la banque KEGG [Kanehisa *et al.*, 2002] a été manuellement traité et plus de la moitié des arêtes du graphe métabolique original a été supprimée.

Il est également possible lors de la construction de considérer la réversibilité des réactions pour orienter certaines arêtes (qui deviennent des arcs). Toujours dans [Ma and An-Ping, 2003], cette information a été manuellement déduite pour chaque réaction et près de la moitié des arêtes a pu être orientée de façon unique (les réactions correspondantes ont été qualifiées d'irréversibles).

Le réseau peut être restreint à un organisme dont le génome est complètement séquencé en ne conservant que les réactions catalysées dans cet organisme. Généralement, c'est sur la base des séquences des gènes prédits et de leur comparaison avec des séquences connues que la présence des catalyseurs (enzymes) est prédite dans un organisme (voir § 4.1). Ce type de données est disponible pour tous les génomes complètement séquencés dans la banque KEGG [Kanehisa *et al.*, 2002] par exemple.

Il est clair que le type de graphe choisi pour représenter les réseaux métaboliques peut avoir une influence importante sur les résultats des analyses et il faut en tenir compte lors de l'interprétation de ces résultats.

Si l'utilisation des graphes se prête assez bien à la représentation des réseaux métaboliques, d'autres formes de représentation, plus complexes, permettent à la fois de représenter les réseaux et de les simuler. C'est le cas des réseaux de Petri.

3.2 Construction d'un réseau de Petri à partir d'un ensemble de réactions

Le paragraphe suivant, introduisant les réseaux de Petri, est basé sur [Valette, 2000].

3.2.1 Présentation des réseaux de Petri

Définition **1** *Réseau de Petri*
Un réseau de Petri est un quadruplet $R = (P, T, Pre, Post)$ *où :*

- P *un ensemble fini de places*
- T *un ensemble fini de transitions*
- $Pre : P \times T \rightarrow \mathbb{N}$, *est l'application places précédentes*
- $Post : T \times P \rightarrow \mathbb{N}$, *est l'application places suivantes*

on note $I(t)$,$t \in T$ *l'ensemble des places* p *telles que* $Pre(p, t)$ *est non nulle*
on note $O(t)$,$t \in T$ *l'ensemble des places* p *telles que* $Post(p, t)$ *est non nulle*

Définition **2** *Réseau marqué*
Un réseau marqué est un couple $N = (R, M)$ *où :*

- $R = (P, T, Pre, Post)$ *est un réseau de Petri*
- $M : P \rightarrow \mathbb{N}$ *est un marquage,* $M(p)$ *est le nombre de jetons contenus dans la place* p

Définition **3** *Tir d'une transition*
Soit un réseau de Petri $R = (P, T, Pre, Post)$*, un marquage* M *et une transition* $t \in T$*, le tir de* t *donne un nouveau marquage* M' *tel que :*

$$\forall p \in P, M'(p) = M(p) - Pre(p, t) + Post(p, t)$$

3.2.1.1 Notation matricielle

Tout marquage M dans un réseau de Petri $R = (P, T, Pre, Post)$ peut être représenté par un vecteur de taille $|P|$. Pre et $Post$ peuvent être représentées sous forme de matrice dont le nombre de lignes est égal au nombre de places et le nombre de colonnes est égal au nombre de transitions. La matrice $C = Post - Pre$ est appelée la matrice d'incidence du réseau.

On note $C(., t)$ la colonne de C associée à la transition t et $C(p, .)$ la ligne de C associée à la place p.

3.2.1.2 Graphe associé

A un réseau de Petri on peut associer un graphe qui possède deux types de nœuds : les places et les transitions. Un arc relie une place p à une transition t si et seulement si $Pre(p, t) \neq 0$. Un arc relie une transition t à une place p si et seulement si $Post(p, t) \neq 0$. Les valeurs non nulles des matrices Pre et $Post$ sont associées aux arcs comme étiquettes (on considère que l'absence d'étiquette correspond à une valuation de 1).

3.2.2 Réseau de Petri et réseaux métaboliques

Un ensemble de réactions peut naturellement être décrit par un réseau de Petri. En effet, si on associe les places aux métabolites et les transitions aux réactions, alors la matrice décrivant l'ensemble des réactions et exactement celle qui décrit le réseau de Petri. Dans le cas des réseaux de Petri, les réactions réversibles doivent être doublées.

Exemple : l'ensemble des trois réactions {Gpm : 3PG $\rightleftharpoons$ 2PG, Eno : 2PG $\rightleftharpoons$ PEP et Pyk : PEP + ADP $\rightarrow$ Pyr + ATP} a pour réseau de Petri équivalent le réseau représenté sur la figure 3.3.

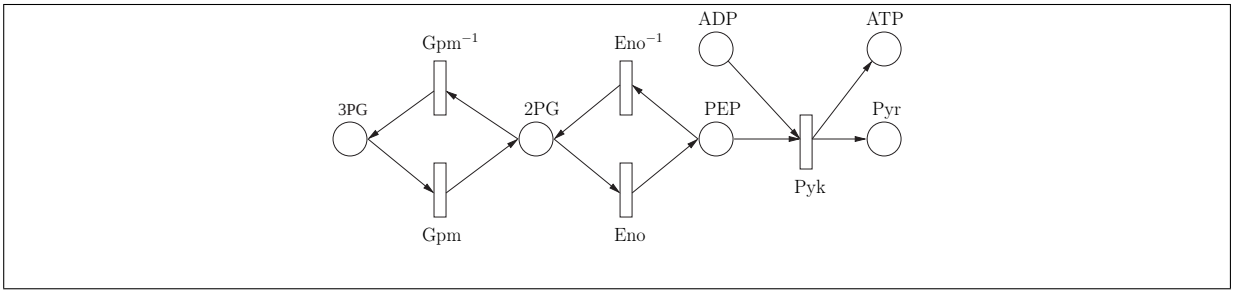


FIG. 3.3: Réseau de Petri pour un ensemble de trois réactions dont deux réversibles

Un intérêt des réseaux de Petri est qu'ils permettent d'intégrer facilement d'autres ressources associées à une réaction comme les catalyseurs par exemple (voir figure 3.4).

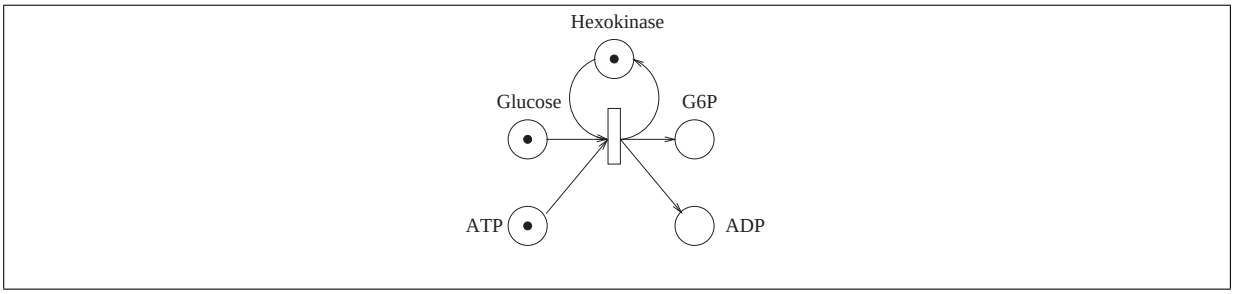


FIG. 3.4: Utilisation d'une ressource supplémentaire pour représenter le catalyseur Hexokinase (une enzyme) dans un réseau de Petri (adapté de [Reddy *et al.*, 1996])

Il est également possible de définir des règles de réécriture qui permettent de simplifier les réseaux. Sur la figure 3.5, le réseau de Petri modélise le cas où une enzyme est représentée par deux états, "actif" ou "inactif". C'est la présence d'un activateur qui permet le passage de l'état "inactif" à l'état "actif". Il est possible pour une enzyme dans l'état "actif" de retrouver l'état "inactif", dans ce cas, la ressource correspondant à l"activateur" est relâchée.

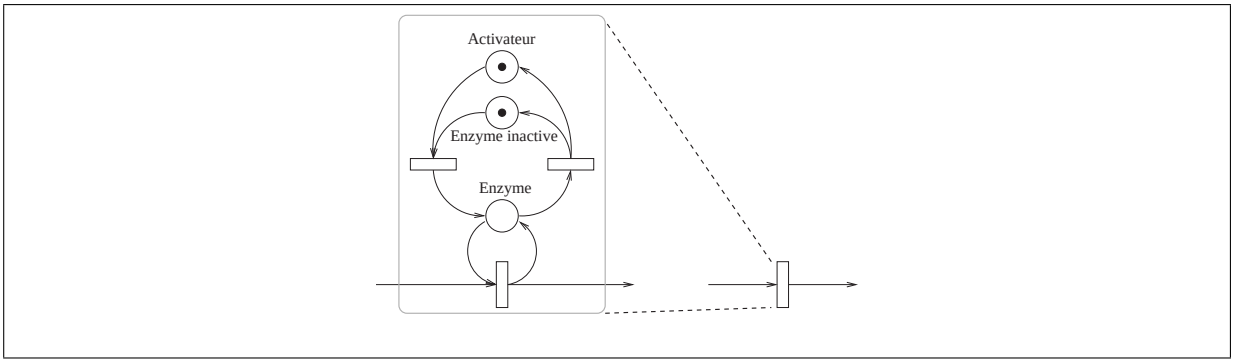


FIG. 3.5: Exemple de simplification dans un réseau de Petri (adapté de [Reddy *et al.*, 1996])

Dans le cas où la présence des ressources "enzyme inactive" et "activateur" est toujours

vérifiée, ce réseau peut être simplifié car il est toujours possible d'obtenir dans cette configuration la ressource "enzyme". Avec la ressource "enzyme", la transition la plus basse peut être tirée et comme les ressources "activateur", "enzyme inactive" et "enzyme" ne servent à aucune autre transition, il est possible de réduire le réseau en ne conservant que la transition.

Ces raisons font que les réseaux de Petri sont utilisés tant pour la représentation que pour la simulation des réseaux métaboliques. Les premiers travaux sur l'utilisation des réseaux de Petri pour la représentation et la manipulation des réseaux métaboliques remontent à [Reddy *et al.*, 1996; 1993]. Depuis, de nombreux autres travaux qui utilisent les réseaux de Petri pour modéliser les réseaux métaboliques sont apparus [Genrich *et al.*, 2001; Heiner *et al.*, 2001]. Un numéro spécial de la revue *In Silico Biology* ([Hofestädt, 2003]) intitulé "Petri Nets for Metabolic Networks" a d'ailleurs été totalement consacré à ce sujet. De plus il existe une plate forme dédiée à la modélisation et à la simulation des réseaux biologiques à l'aide des réseaux de Petri hybrides nommée *GenomicObject-Net* [Matsuno *et al.*, 2003].

Il y a deux manières d'aborder le problème de la reconstruction de voies métaboliques. Ces deux approches sont fondamentalement différentes :

- la reconstruction par homologie s'appuie sur la connaissance de voies métaboliques connues pour effectuer la reconstruction en tentant d'adapter ces voies à un nouvel organisme
- la reconstruction *ab initio* ne nécessite aucune connaissance *a priori* sur la voie à reconstruire mais se base sur la connaissance de l'ensemble des réactions qui peut potentiellement la constituer

Chapitre 4

La reconstruction par homologie

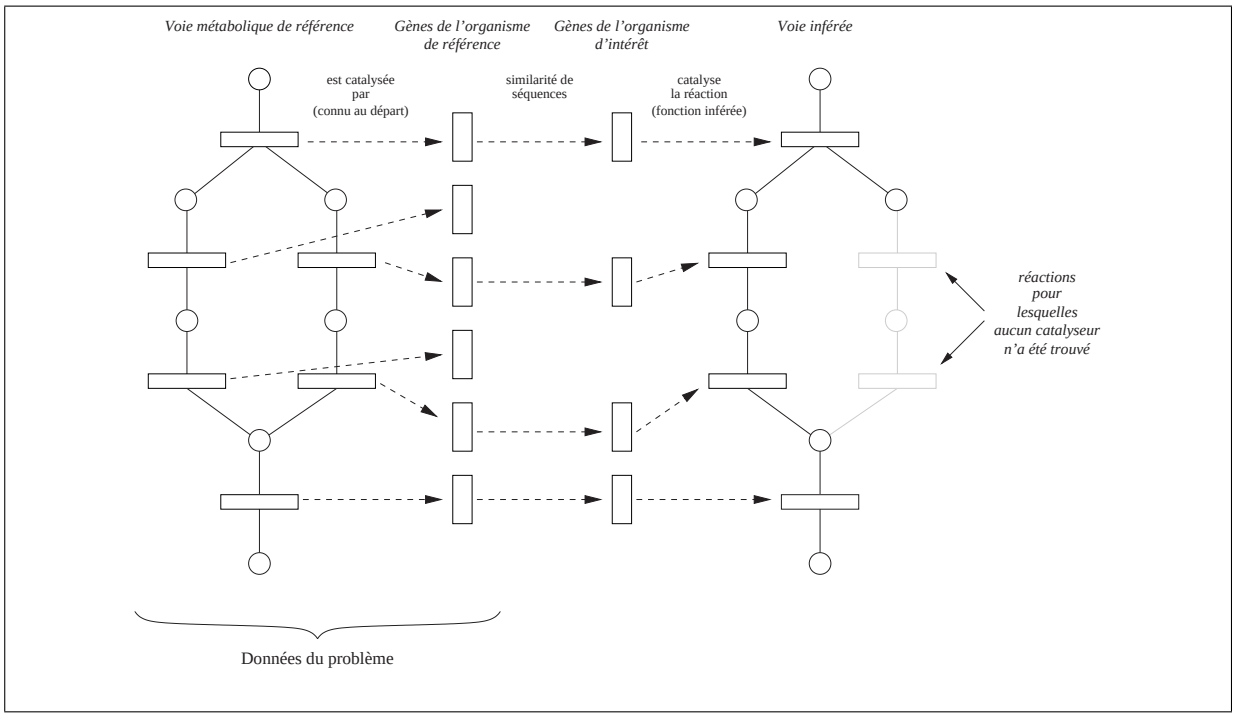


Fig. 4.1: Principe de la reconstruction de voie métabolique par homologie - La voie métabolique à reconstruire est supposée entièrement connue dans l'organisme de référence, la similarité de séquences est utilisée afin de trouver dans l'organisme cible les gènes responsables de la catalyse des réactions impliquées dans la voie. Si le nombre de catalyseurs trouvés est suffisant, la voie est alors considérée comme présente dans l'organisme cible

La reconstruction par homologie est le type de reconstruction automatique le plus utilisé. Elle s'appuie sur la connaissance d'un ensemble de voies métaboliques et vérifie, pour chaque voie, si elle est présente dans l'organisme étudié. Son principe de fonctionnement est illustré sur la figure 4.1.

On peut scinder la résolution du problème de la reconstruction par homologie en deux étapes successives :

1. il faut assigner pour chaque gène de l'organisme cible une, aucune ou plusieurs fonctions enzymatiques (le plus souvent, ces fonctions sont assignées automatiquement sur la base de la similarité de séquences)
2. en s'appuyant sur un recueil de voies métaboliques connues, il faut confronter chaque voie métabolique avec les fonctions enzymatiques prédites pour l'organisme cible (prédictions faites pour les gènes)

Ces deux points sont successivement abordés dans les paragraphes suivants.

4.1 Assignation des fonctions enzymatiques

L'inférence, à grande échelle, de la fonction du produit des gènes est un problème fréquemment rencontré, surtout depuis que les programmes de séquençage de génome se multiplient. En effet, l'étude de la séquence complète d'un génome permet d'identifier des gènes (assez efficacement pour les génomes procaryotes et avec plus de difficultés pour les génomes eucaryotes). Chaque fois qu'un nouveau génome est séquencé, il faut prédire la fonction de plusieurs milliers de gènes.

Le plus souvent, la seule information sur laquelle se base cette prédiction est la séquence protéique obtenue par la traduction de la séquence du gène (mais on peut également, pour inférer la fonction, s'aider de la proximité chromosomique d'autres gènes dont la fonction est connue [Huynen *et al.*, 2000; Overbeek *et al.*, 1999a]). Ce type de prédiction fonctionnelle repose sur des hypothèses qui vont être explicitées dans le prochain paragraphe.

Deux types d'approches sont utilisés pour la prédiction fonctionnelle des protéines qui sont développés dans les paragraphes suivants :

- à partir de familles de séquences complètes dont les protéines correspondantes partagent la même fonction
- à partir de signatures (sur la séquence) caractéristiques de fonctions spécifiques

4.1.1 Homologie, orthologie et paralogie, fonction et similarité entre séquences

On dit de deux gènes qu'ils sont *homologues* lorsqu'ils dérivent d'un gène ancestral commun. Introduites par [Fitch, 1970], les relations d'*orthologie* et de *paralogie* décrivent plus précisément la relation entre deux gènes homologues. Ce qui permet de distinguer ces deux cas est la connaissance des événements de spéciation et de duplication. Si l'événement

le plus proche qui sépare les deux gènes est une spéciation, alors les deux gènes sont dits *orthologues*, si au contraire, l'événement le plus proche qui sépare les deux gènes est une duplication, alors les deux gènes sont dits *paralogues*.

Ces deux définitions sont illustrées sur la figure 4.2. Depuis le sommet de l'arbre, qui représente le gène ancestral, il y a eu un événement de spéciation, puis, sur la branche de droite, une duplication, suivie à nouveau d'une spéciation (représentée par deux ronds), puis pour la branche la plus à droite à nouveau un événement de duplication. Cela donne 3 espèces différentes H, R et S car il y a eu deux événements de spéciation. Avec ce scénario, l'espèce H a une copie du gène, l'espèce R en a deux tandis que l'espèce S en a trois. Les gènes R1 et S1 sont orthologues, ainsi que R2 et S2 ou R2 et S3, tandis que S2 et S3 sont paralogues, au même titre que R1 et S2 ou R1 et S3.

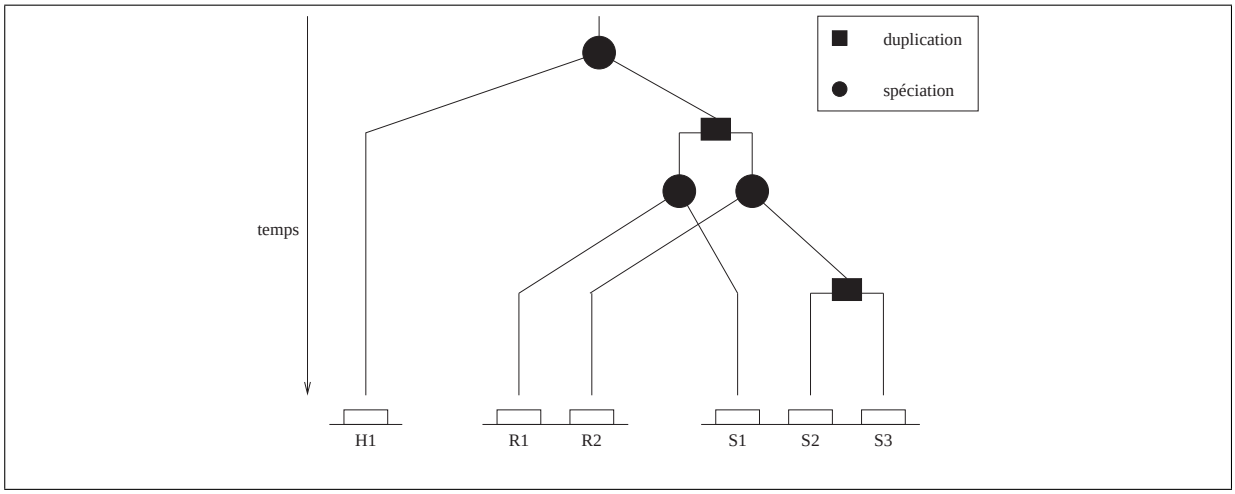


FIG. 4.2: Illustration des relations d'homologie, orthologie et paralogie (adapté de [Fitch, 2000])

La figure 4.2 permet de mettre en évidence deux propriétés importantes des relations d'orthologie et de paralogie :

i) elles ne sont pas transitives (S2 est orthologue à R2, R2 est orthologue à S3 mais S2 et S3 ne sont pas orthologues)

ii) elles ne sont pas bijectives (R2 est orthologue à S2 et S3, H1 est orthologue à tous les Ri et Si)

La définition de la relation d'orthologie est donc de nature purement phylogénétique et n'a *a priori* rien à voir avec la fonction des gènes. De nombreux auteurs établissent néanmoins un lien entre l'orthologie et la fonction. Ce lien repose sur les hypothèses suivantes :

– après un événement de spéciation, les deux copies du gène (une dans chaque espèce) conservent la même fonction

– après un événement de duplication, une des deux copies du gène (dans la même espèce) conserve la fonction initiale du gène ancestral et l'autre copie peut évoluer vers une fonction similaire ou différente

Dans ces conditions, on peut inférer que deux gènes orthologues ont de fortes chances de présenter la même fonction (la réciproque est fausse).

L'étape suivante consiste à établir l'orthologie ou la paralogie d'un couple de gènes. En théorie, cette relation devrait être établie sur la base d'une reconstruction phylogénétique. Malheureusement, il est rare de procéder ainsi et la plupart des auteurs se base essentiellement sur la similarité de séquences en faisant l'hypothèse que si les deux séquences sont "suffisamment" similaires, alors elles ont de fortes chances d'être orthologues.

On constate que ce "raccourci" est souvent abusif. La connaissance de génomes complets permet néanmoins d'atténuer un peu cette hypothèse. Par exemple, [Tatusov *et al.*, 1997] introduit la notion de *Bidirectional Best Hit* (BBH) : étant donné un gène a dans un génome $\mathcal{A}$, si parmi tous les gènes d'un génome $\mathcal{B}$, le gène b est le plus ressemblant à a et si, inversement, parmi tous les gènes de $\mathcal{A}$, a est le plus ressemblant à b alors on admet que a et b sont orthologues (et donc partagent probablement la même fonction).

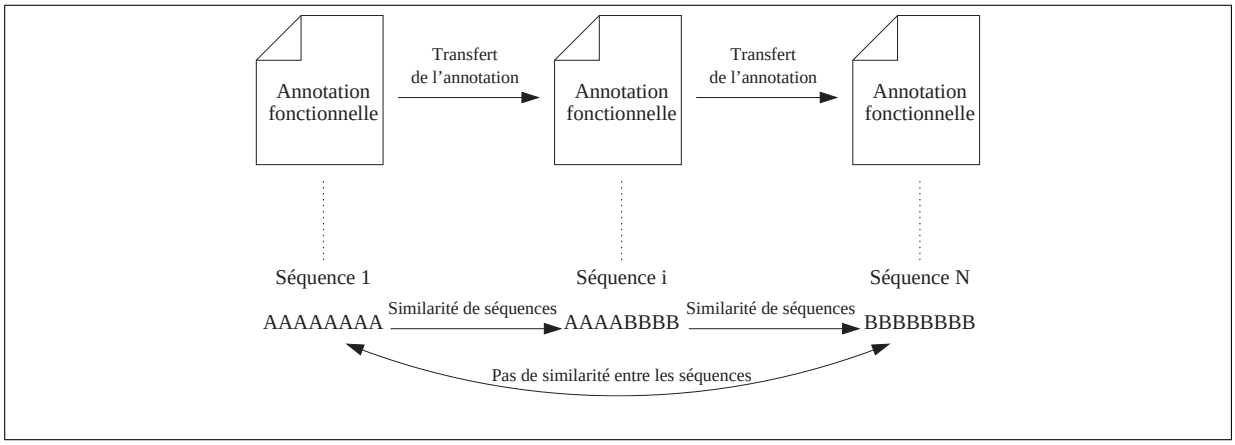


Fig. 4.3: Le transfert systématique d'annotations fonctionnelles sur la base de similarité entre séquences peut provoquer des erreurs d'annotation

Une première façon d'attribuer une fonction à un gène est donc de rechercher des similarités entre sa séquence (ou celle de son produit) et les séquences des gènes contenus dans le génome d'un organisme proche et déjà caractérisé [Tatusov *et al.*, 1996]. Cette solution n'est cependant pas complètement satisfaisante. Même entre organismes proches, les fonctions enzymatiques représentées peuvent ne pas être les mêmes [Cordwell, 1999], ou encore, deux enzymes ayant la même fonction peuvent ne partager aucune similarité de séquences [Galperin *et al.*, 1998]. Transférer les annotations fonctionnelles sur la base d'une similarité entre deux séquences peut amener à des erreurs d'annotations. Par ailleurs

une telle stratégie d'annotation, risque de propager des erreurs d'annotation. Cela vient du fait que la similarité de séquence n'est pas une relation transitive (voir figure 4.3).

Il faut donc concevoir des méthodes plus robustes pour la prédiction des fonctions enzymatiques que la simple recherche de séquences similaires. Un moyen d'accroître la qualité de la prédiction est de se baser non plus sur la similarité avec une seule séquence mais sur la similarité avec un groupe de séquences supposées orthologues (§ 4.1.2). Il est également possible de caractériser ces familles de séquences par des signatures (§ 4.1.3). C'est alors sur l'occurrence dans la séquence candidate de signatures spécifiques que se base l'annotation.

4.1.2 Utilisation de familles de séquences orthologues

Un moyen de contourner le problème du transfert erroné d'annotation par la similarité entre deux séquences est de construire des familles de séquences orthologues. Les séquences de chaque famille étant orthologues, il est supposé (voir § précédent) que toutes les protéines associées à ces séquences partagent la même fonction. On peut alors associer à la famille l'annotation fonctionnelle consensuelle de ses membres. Si une nouvelle séquence est classée comme faisant partie d'une famille déjà définie (elle est donc considérée comme orthologue à toutes les séquences de la famille), c'est un indice suffisant pour lui assigner la même annotation fonctionnelle que celle associée à la famille. La décision d'associer l'annotation fonctionnelle à la séquence se base ainsi non plus sur la similarité avec une seule séquence mais sur la similarité avec l'ensemble des séquences membres de la famille.

4.1.2.1 Construction de familles de séquences orthologues

Il existe plusieurs façons de construire des familles de séquences orthologues, mais la première étape consiste toujours à comparer deux à deux l'intégralité des séquences que l'on veut classer pour en inférer des liens d'orthologie entre chaque couple de séquences. Cette comparaison deux à deux est effectuée le plus souvent en calculant la distance d'édition entre les deux séquences.

Définition **4** *Distance d'édition*
Soient deux chaînes de caractères s_1 et s_2 définies sur un alphabet commun Σ et un ensemble de transformations élémentaires (la délétion, l'insertion et la substitution d'un caractère) auxquelles est associé un coût individuel, la distance d'édition entre les deux chaînes s_1 et s_2 est le coût minimum des transformations élémentaires à appliquer à la chaîne s_1 pour obtenir la chaîne s_2

Exemple : la figure 4.4 montre quelques un des chemins les plus courts permettant de passer de la chaîne $s_1 = aagcg$ à la chaîne $s_2 = agagt$.

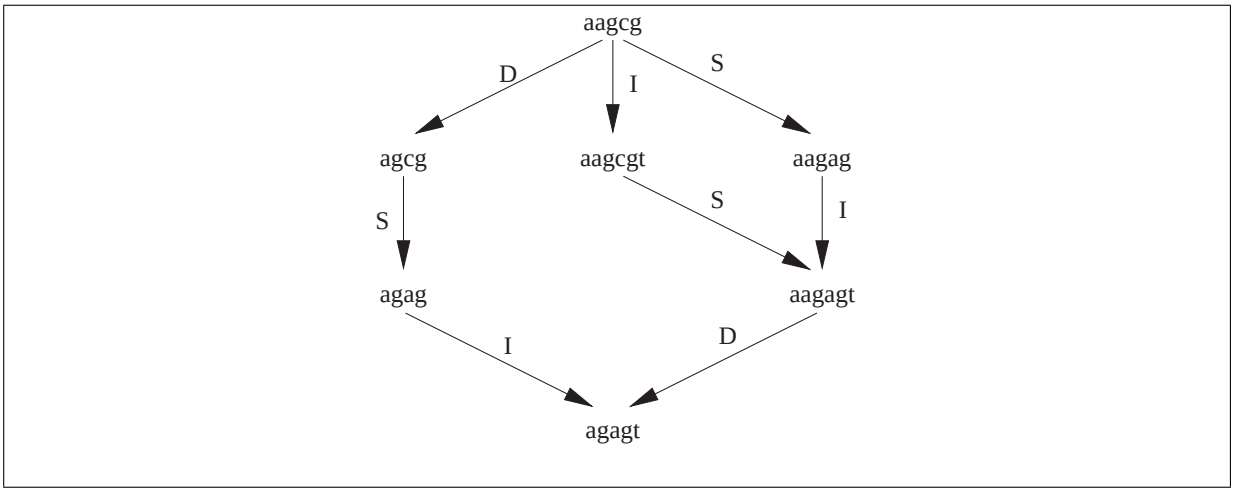


Fig. 4.4: Distance d'édition entre deux séquences - Dans le cas où les trois opérations élémentaires délétion(D), insertion(I) et substitution(S) ont un coût identique de 1, alors la distance d'édition entre les deux chaînes $s_1 = aagcg$ et $s_2 = agagt$ est de 3 (c'est le plus petit nombre d'opérations à réaliser pour passer de s_1 à s_2)

Le critère pour décider s'il y a un lien d'orthologie entre deux séquences s_1 et s_2 de deux organismes $\mathcal{G}_1$ et $\mathcal{G}_2$ est généralement un des trois critères suivants (ou une association de ces critères) :

1. la distance d'édition entre s_1 et s_2 est au dessous d'un seuil fixé
2. s_2 est la séquence de $\mathcal{G}_2$ qui est la plus proche de s_1
3. s_1 est la séquence de $\mathcal{G}_1$ qui est la plus proche de s_2

La combinaison de 2 et 3 correspond au critère BBH (*Bidirectional Best Hit*) mentionné au § 4.1.1.

Il faut noter que sur la base du critère BBH, la relation inférée n'est alors pas réflexive. Si le critère BBH est utilisé, dans le cas de duplications récentes d'une des deux séquences, il est possible que le lien soit raté.

Sur la base d'une telle relation d'orthologie, il est possible de définir des familles de séquences orthologues. Dans le meilleur des cas, toutes les séquences regroupées au sein d'une même famille devraient être orthologues deux à deux.

Il est possible de représenter ces relations d'homologie dans un graphe où chaque nœud représente une séquence et où chaque arête entre deux nœuds indique que les deux séquences associées à ces deux nœuds sont homologues. Dans un tel graphe, une clique représente un groupe de séquences toutes deux à deux homologues au sens d'un des critères

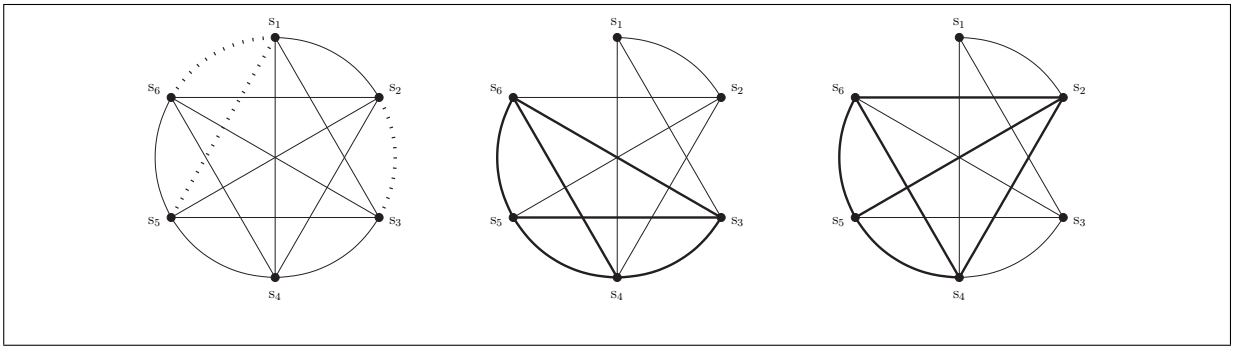


Fig. 4.5: Effet de la perte de quelques arêtes sur la taille de la clique de taille maximale dans un graphe complet

précédents et devant donc appartenir à une même famille. Dans ce graphe, les familles correspondent donc idéalement aux cliques de taille maximale. Dans la réalité, la taille des cliques est souvent réduite et leur nombre important car certaines relations d'homologie ne sont pas trouvées [Bize *et al.*, 2001].

Exemple : le graphe de la figure 4.5 montre un graphe complet représentant un ensemble de 6 séquences homologues deux à deux. Si trois liens d'homologie (sur quinze possibles) sont ratés, la taille maximale de la famille qu'il est possible de construire passe de six à quatre. Le nombre de familles de taille maximale passe de 1 à 2 et 3 séquences sont communes à ces deux familles (s_4, s_5 et s_6).

Comme la recherche de clique de taille maximale dans un graphe est un problème difficile [Garey and Johnson, 1979] et que les graphes représentant les relations d'homologie entre séquences peuvent être de taille importante, de nombreuses heuristiques sont mises en œuvre. L'utilisation des heuristiques a pour objectif principal d'augmenter la taille des familles construites tout en garantissant que la ressemblance des séquences reste bonne au sein de la famille. Dans [Perrière *et al.*, 2000], les critères stringents appliqués à la construction des liens d'homologie entre deux séquences autorisent de construire les familles en suivant les liens d'homologie de façon transitive (*i.e* deux séquences d'une même famille peuvent être reliées indirectement par des liens d'homologie). Chaque composante connexe du graphe représente alors une famille de séquences homologues. Dans [Tatusov *et al.*, 1997], on commence par construire toutes les cliques de taille trois qui représentent des familles initiales. Chaque clique initiale est donc un triangle. Deux familles sont fusionnées pour former une famille de plus grande taille si ces deux familles ont un segment commun, *i.e.* utilisent un même lien d'homologie. [Fujibuchi *et al.*, 2000] introduit la notion de graphe complet à P %, *i.e.* qui contient au moins P % des arcs totaux d'un graphe complet de même taille. Dans ce cas, une famille est un plus grand sous-graphe complet à P %.

4.1.2.2 Evaluation de l'appartenance d'une séquence à une famille de séquences homologues

Une fois que les familles sont disponibles, il est possible, pour une nouvelle séquence, de tester son appartenance aux différentes familles. Dans certains cas, l'ajout d'une nouvelle séquence peut impliquer le recalcul de toutes les familles ou d'un nombre important d'entre elles.

4.1.3 Utilisation de signatures spécifiques de fonctions

Un autre moyen de procéder à l'assignation de fonction est la recherche dans la séquence de signatures spécifiques à une fonction. Les protéines ayant une même fonction partagent souvent, au niveau de leur séquence, des caractéristiques de cette fonction appelées signatures. L'occurrence dans une séquence protéique candidate d'une signature conservée dans un groupe de séquences associées à une même fonction est un bon indice pour associer la même fonction à la séquence candidate. Il existe plusieurs façons de définir des signatures et de les construire. L'évaluation dépend bien entendu du type de signature utilisé.

4.1.3.1 Définition des signatures

Il existe de nombreuses façons de décrire une signature. Les plus utilisées sont :
- les expressions de type expressions régulières
- les tableaux poids-positions
- les modèles de Markov à états cachés de type "profil" (profile HMM)

Expressions de type expressions régulières Les expressions de type expressions régulières (qui décrivent un langage régulier) sont capables de décrire la plupart des signatures rencontrées dans les séquences biologiques. Il existe plusieurs variantes de ce type d'expressions, mais la plus utilisée semble être les expressions de type "network expression" [Brazma *et al.*, 1998]. Les "network expression" sont des expressions régulières sans la fermeture de Kleene (les répétitions non-bornées ne sont pas autorisées) et où la disjonction ne s'exprime qu'au niveau d'une porsition.

Exemple : l'expression suivante est la signature des enzymes de type pyruvate kinase : [LIVAC]- ?-[LIVM](2)-[SAPCV]-K-[LIV]-E-[NKRST]- ?-[DEQHS]-[GSTA]-[LIVM]. Ici, l'alphabet utilisé est l'alphabet à 21 lettres représentant les acides aminés et le joker est ' ?'. Les crochets indiquent qu'un choix est possible entre les caractères contenues entre les crochets. Les nombres entre parenthèses indiquent une répétition du motif les précédant.

Tableaux poids-positions Un tableau poids-positions est un tableau qui représente la fréquence de chaque caractère pour chaque position dans la signature.

Exemple :soient trois occurrences :

$$\begin{array}{rcccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ o_1 = & a & b & b & a & b & c \\ o_2 = & a & b & a & b & a & b \\ o_3 = & a & b & a & b & b & b \end{array}$$

d'un même motif de longueur 6 sur un alphabet $\Sigma = \{a, b, c, d\}$. Le tableau poids-position correspondant à la description du motif est le suivant :

positions

	1	2	3	4	5	6
a	1	0	2/3	1/3	1/3	0
b	0	1	1/3	2/3	2/3	2/3
c	0	0	0	0	0	1/3
d	0	0	0	0	0	0

Ce modèle a la particularité de faire l'hypothèse d'indépendance des positions et de ne pas prendre en compte les insertions et les délétions. Le modèle de base a donc été étendu [Gribskov *et al.*, 1988] afin de prendre en compte un coût d'insertion/délétion à chaque position du tableau ce qui donne un modèle de Markov à états cachés très simple (les états sont caractérisés par une chaîne de Markov d'ordre 0).

Modèles de Markov à états cachés de type "profil" Un modèle de Markov à états cachés est un automate stochastique [Eddy, 1998]. A chacune des transitions entre deux états est associée la probabilité d'effectuer la transition entre les deux états. Par définition, la somme des probabilités quittant un état est égale à 1. A chaque état sont associées autant de probabilités que la taille de l'alphabet considéré, la somme de ces probabilité est égale à 1. Ces probabilités sont celles des émissions de chaque symbole dans cet état. On peut également introduire dans un modèle de Markov à états cachés des états muets (qui ne sont associés à l'émission d'aucun symbole).

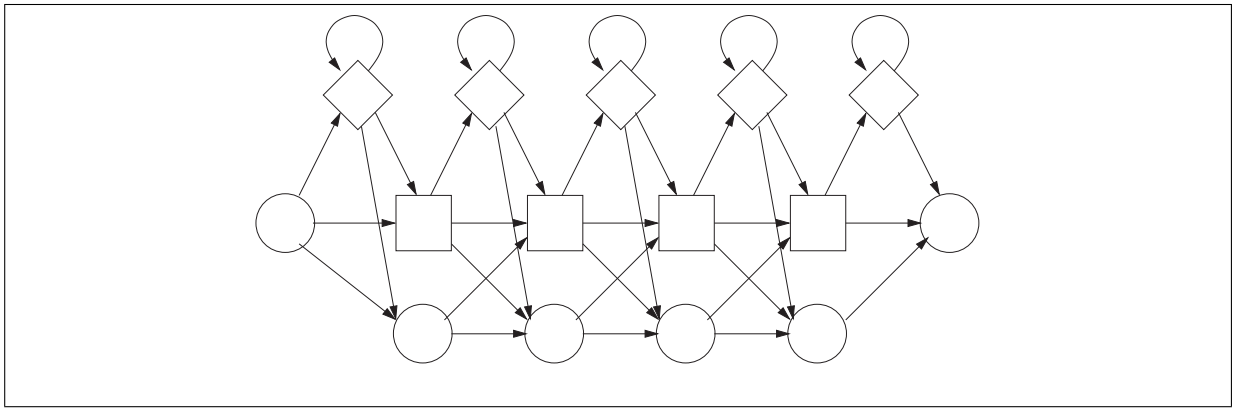


Fig. 4.6: Représentation d'un profil à l'aide d'un modèle de Markov à états cachés - Les carrés représentent des états émetteurs qui correspondent à l'émission d'un caractère faisant partie du consensus de la signature, les losanges représentent des états d'insertions et les ronds représentent des états muets qui correspondent à des délétions dans la signature

Pour modéliser les signatures dans les séquences, on utilise des modèles de Markov à états cachés caractérisés par une topologie bien particulière (voir la figure 4.6).

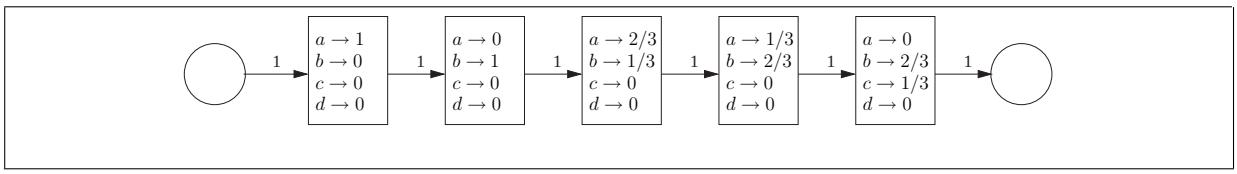


Fig. 4.7: Représentation d'un tableau poids-position à l'aide d'un modèle de Markov à états cachés

Il est très facile de transformer un tableau poids-positions en modèle de Markov à états cachés de type profil en créant autant d'états que de colonnes dans le tableau. Les probabilités d'émission des symboles sont les mêmes que celles contenues dans chaque colonne du tableau. On associe aux probabilités de transitions entre les états la probabilité 1. La figure 4.7 montre le modèle de Markov à états cachés correspondant au tableau poids position de l'exemple précédent.

4.1.3.2 Inférence et construction des signatures

L'inférence d'expressions de type "network expression" à partir d'un ensemble de séquences non alignées est un problème algorithmique qui donne lieu à de nombreuses recherches [Pisanti and Sagot, 2003].

Pour l'inférence de tableaux poids-positions, il existe des algorithmes permettant d'apprendre, à partir de séquences supposées contenir la même signature, les valeurs des probabilités d'émission de chaque symbole [Lawrence *et al.*, 1993].

De la même façon, il existe des algorithmes pour estimer la valeurs des probabilités des modèles de Markov à états cachés à partir d'exemples [Durbin *et al.*, 1998]. Mais la manière la plus utilisée pour paramétrer ces deux derniers types de modèles semble l'utilisation d'alignements multiples déjà constitués qui fournissent une meilleure information. Il est également possible de constituer les alignements de manière itérative en même temps que les signatures grâce à des logiciels comme PSI-Blast [Altshull *et al.*, 1997]. Si les séquences sont groupées par leur fonction enzymatique, il est alors possible de déduire des signatures spécifiques à chaque fonction enzymatique (comme dans le cas de [Renard-Claudel *et al.*, 2001; 2003]).

4.1.3.3 Evaluation de l'occurrence d'une signature dans une séquence

Pour le cas des expressions régulières, il faut souvent tester non pas l'occurrence stricte de la signature mais plutôt si la signature apparaît de manière approximative. Il existe des algorithmes performants pour effectuer de telles recherches [Myers, 1996].

Dans le cas des tableaux poids-positions et des modèles de Markov à états cachés, le résultat de l'évaluation du modèle sur une séquence candidate est un score (ou une série de score) ou une probabilité. Dans les deux cas, il faut fixer un seuil limite au-delà duquel on considère qu'il y a occurrence ou non de la signature. Dans les deux cas, cela demande une étape de calibration utilisant des ensembles de tests positifs et négatifs.

Il existe de nombreuses banques de données de signatures extraites automatiquement ou expertisées manuellement dont [Corpet *et al.*, 2000; Haft *et al.*, 2001; Falquet *et al.*, 2002; Bateman *et al.*, 2002; Gattiker *et al.*, 2003; Marchler-Bauer *et al.*, 2003; Mulder *et al.*, 2003]. La plupart des signatures sont fournies sous la forme de tableaux poids-positions ou de modèles de Markov à états cachés.

Même si les méthodes pour la prédiction de fonctions enzymatiques semblent donner de bons résultats (voir par exemple [Renard-Claudel *et al.*, 2003]), la prédiction de fonctions enzymatiques pour les protéines reste un problème ouvert car de nombreuses fonctions enzymatiques ne sont que peu ou pas représentées dans les banques de séquences spécialisées. Ce problème est d'autant plus important que la prédiction des fonctions enzymatiques d'un organisme est une condition essentielle pour pouvoir effectuer la reconstruction des voies métaboliques de cet organisme de façon satisfaisante.

4.2 Méthode de reconstruction des voies métaboliques

La plupart des systèmes de reconstruction automatique de voies métaboliques [Gaasterland and Selkov, 1995; Bansal, 2000; 2001; Overbeek *et al.*, 2000; 1999b; Karp *et al.*, 1999; 2002a] utilisent le même principe, déjà illustré sur la figure 4.1.

La première étape consiste à identifier les fonctions enzymatiques présentes dans l'organisme étudié sur la base de la séquence complète de son génome. Suivant les systèmes, cela est mis en œuvre soit grâce à la similarité de séquences [Gaasterland and Selkov, 1995; Bansal, 2000; 2001; Overbeek *et al.*, 2000; 1999b] soit à partir des annotations fournies avec la séquence complète du génome [Karp *et al.*, 1999; 2002a]. Dans le premier cas, les séquences des gènes prédits sont comparées à une banque de référence contenant des séquences d'enzymes bien caractérisées. Dans le second cas, cette information est extraite à partir des annotations fournies avec la séquence complète du génome.

La seconde étape consiste à prédire effectivement les voies métaboliques supposées présentes dans l'organisme. Pour cela une base de données contenant l'intégralité des voies métaboliques connues pour des organismes de référence est utilisée.

Suivant les enzymes qui ont été prédites pour l'organisme d'intérêt les voies transposées pourront être incomplètes.

Exemple : la figure 4.8 montre une transposition de la voie de biosynthèse du tryptophane de *Escherichia coli* à *Hæmophilus influenzae* extraite de l'outil BioCyc [Karp *et al.*, 2000]. Pour deux des cinq étapes de la voie transposée, aucune enzyme n'a pu être prédite dans *Hæmophilus influenzae*, alors que pour les trois autres étapes, au moins un gène a été trouvé pour la protéine responsable de la catalyse de cette étape.

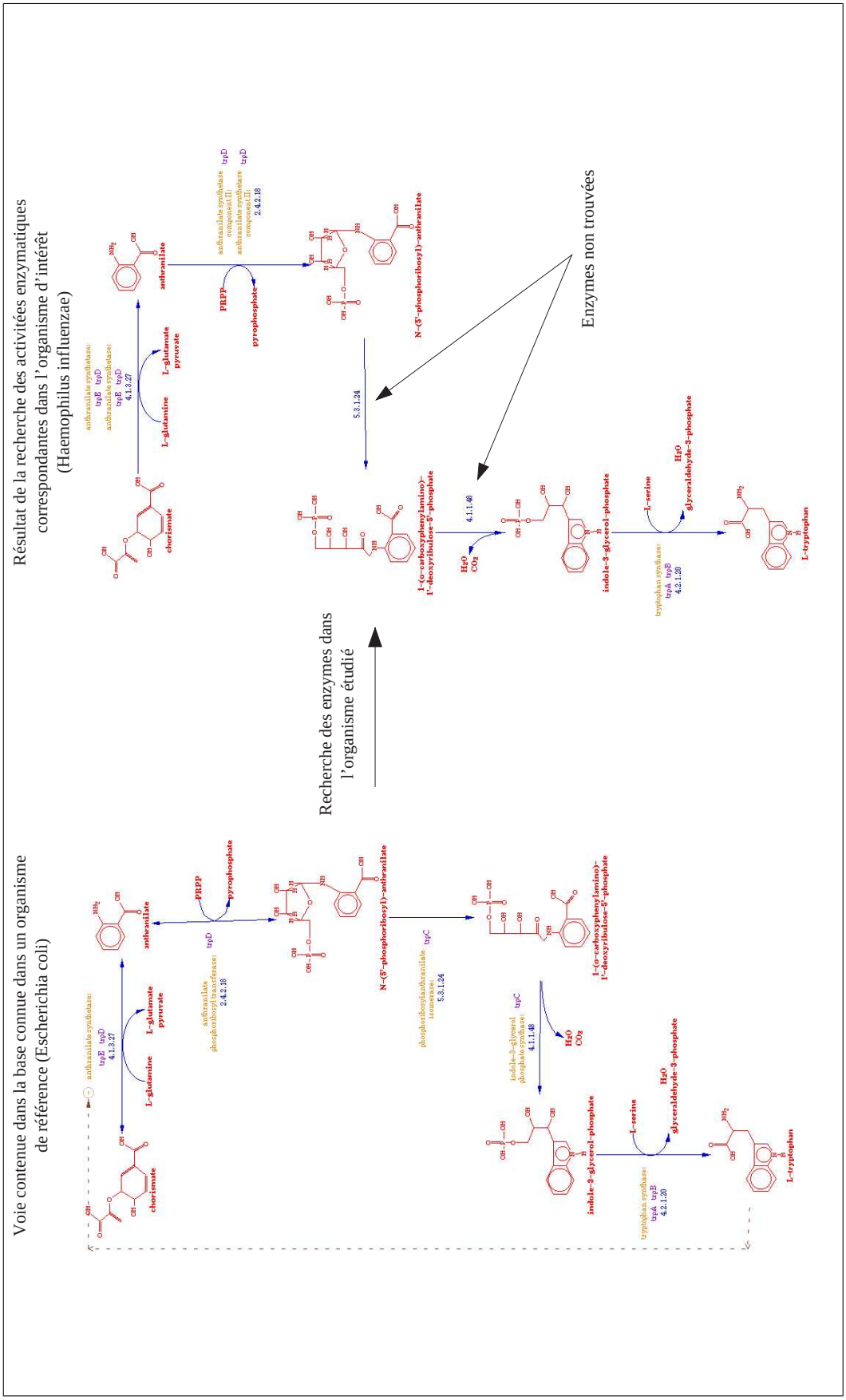


FIG. 4.8: Transposition de la voie de biosynthèse du tryptophane d'*Escherichia coli* à *Hæmophilus influenzae* - dans ce cas précis, l'occurrence de la voie d'*Escherichia coli* dans *Hæmophilus influenzae* est partielle (extrait de l'outil BioCyc [Karp *et al.*, 2000])

Dans ce cas, il faut décider si oui ou non, la voie doit être conservée. Pour cette prise de décision, [Overbeek *et al.*, 1999b; Karp *et al.*, 2002a] font intervenir le calcul d'un score qui dépend :

- du nombre total d'enzymes impliquées dans la voie
- du nombre total d'enzymes impliquées dans la voie et présentes dans l'organisme
- du nombre d'enzymes effectivement présentes dans l'organisme et utilisées exclusivement dans cette voie
- du nombre d'enzymes présentes dans l'organisme et utilisées également dans d'autres voies

Au dessus d'un certain score, la voie est conservée. Pour l'exemple donné précédemment, bien qu'incomplète la voie a été conservée.

4.3 Evaluation de la reconstruction par homologie

Dans [Paley and Karp, 2002], une évaluation de la reconstruction par homologie a été menée manuellement en comparant, pour la bactérie *Helicobacter pylori*, les prédictions du programme PathoLogic [Karp *et al.*, 2002a] avec les réseaux métaboliques compilés par un expert [Marais *et al.*, 1999]. Il semble que la qualité des prédictions soit tout à fait satisfaisante. Cela vient probablement de la richesse de la base de données de voies métaboliques MetaCyc [Karp *et al.*, 2002b] utilisée pour la phase de prédiction (et aussi de la ressemblance d'*Helicobacter pylori* avec d'autres bactéries pour lesquelles de nombreuses voies métaboliques sont dans cette base).

Cependant, des études ont montrées que, même pour des voies essentielles comme la glycolyse [Dandekar *et al.*, 1999] ou le cycle de Krebs [Huynen *et al.*, 1999], il existe une très grande diversité non seulement pour les catalyseurs utilisés, mais également dans la topologie des voies. Suivant les cas, certaines portions des voies sont incomplètes ou utilisent des chemins alternatifs (voir la figure 3.1(a) pour un exemple de voies alternatives dans deux organismes). Il faut bien être conscient qu'à moins de disposer d'une base de données exhaustive, un tel système ne sera jamais capable de prédire l'intégralité des voies métaboliques d'un organisme. Même si cela n'enlève en rien l'intérêt de disposer d'un tel outil, d'autres méthodes moins dépendantes des connaissances sur les voies métaboliques sont donc nécessaires pour prédire des voies métaboliques inédites ou des voies alternatives. C'est le but des méthodes de reconstruction de voies métaboliques dites *ab initio*.

Chapitre 5

La reconstruction *ab initio*

La reconstruction *ab initio* peut être définie comme la recherche à partir d'un ensemble de réactions et un ensemble de composés d'intérêt, de sous-ensembles de réactions permettant la synthèse de certains de ces composés à partir d'autres. L'ensemble de réactions considéré au départ peut être celui des réactions intervenant dans un contexte particulier, ou l'ensemble des réactions associées à des enzymes connues ou prédites dans un organisme, ou encore l'ensemble de toutes les réactions connues. Différents types d'approches ont été imaginés pour la reconstruction *ab initio* :

- la recherche de chemins dans le graphe des composés
- la recherche d'ensembles de réactions respectant un bilan réactionnel global
- la construction de réseaux des flux de carbone
- la recherche des chemins suivis par les atomes des composés

5.1 Approximation des réactions par des relations binaires et recherche de chemins dans le graphe des composés

Dans le but d'identifier des voies métaboliques ou des voies alternatives, il est possible d'approximer chaque réaction par des relations binaires entre deux composés impliqués dans la réaction, l'un comme substrat et l'autre comme produit [Susumu *et al.*, 1997]. Une telle relation signifie que la réaction transforme le composé substrat en composé produit. Dans le cas où la réaction est réversible, cette relation est doublée.

Exemple : la réaction de la figure 5.1 peut ainsi être approximée par deux relations binaires L-glutamate ↔ L-glutamyl 5-phosphate et ATP ↔ ADP. Aucune relation entre L-glutamate et ADP, ainsi qu'entre L-glutamyl 5-phosphate et ATP n'est extraite car, bien

qu'impliqués dans la même réaction, ces composés n'échangent que très peu d'atomes dans la réaction.

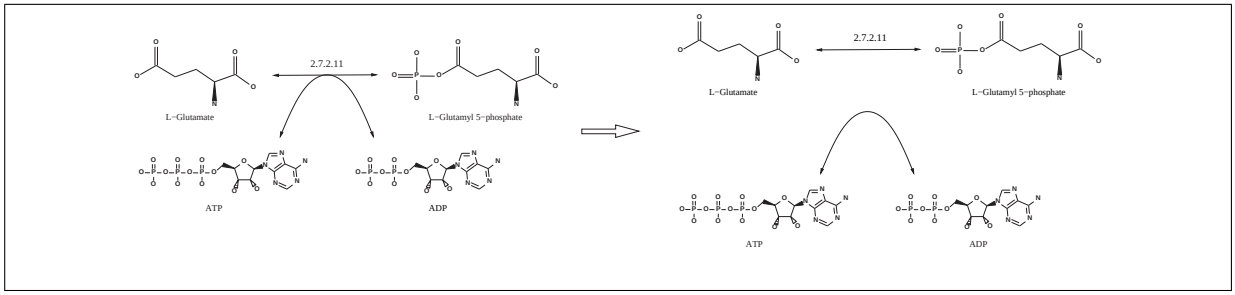


Fig. 5.1: Décomposition d'une réaction en relations binaires

Sur la base de telles relations, il est envisageable d'un point de vue biochimique de considérer qu'une succession de relations entre deux composés correspond à une voie métabolique possible. La recherche de ces successions se ramène alors à la recherche d'un chemin dans le graphe où les composés sont les nœuds du graphe et les arcs la relation binaire.

Cette méthode est très simple à mettre en œuvre. Elle nécessite toutefois une expertise manuelle pour obtenir l'approximation des réactions. Mais son inconvénient majeur vient de l'hypothèse, facilement mise en défaut, que toute réaction peut être décomposée en relations binaires. Cette hypothèse n'est pas toujours vérifiée notamment avec les réactions de condensation (par exemple les lyases) dans lesquels deux réactifs doivent être simultanément associés au même produit. La décomposition binaire ne permet pas de conserver le lien entre les deux réactifs et entraîne l'existence de chemins réactionnels non pertinents.

Exemple : la figure 5.2 illustre le cas où la décomposition en relations binaires d'une réaction ne permet pas de la décrire correctement.

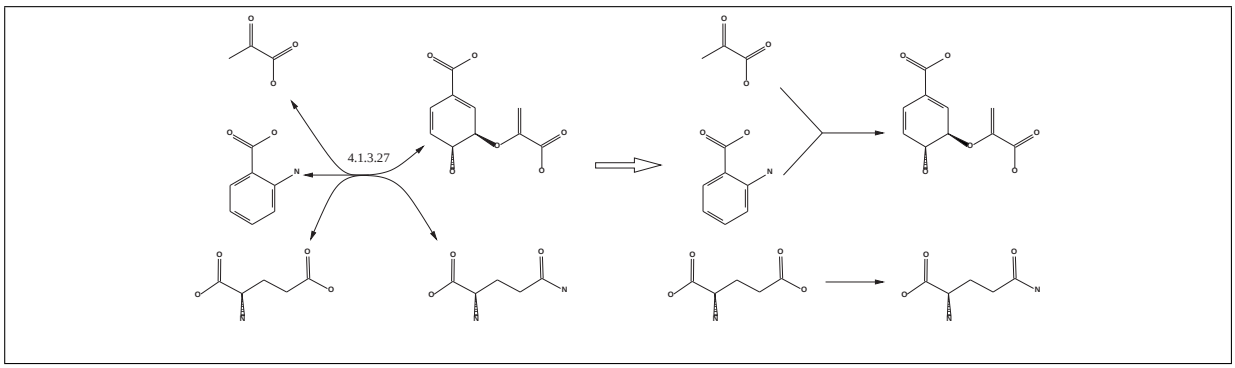


Fig. 5.2: Cas d'une réaction non descriptible par des relations binaires - Dans la réaction de numéro EC 4.1.3.27, qui fait partie de la classe des lyases, la relation entre l'anthranilate, le pyruvate et le chorismate ne peut pas être décrite par un ensemble de relations binaires entre deux composés

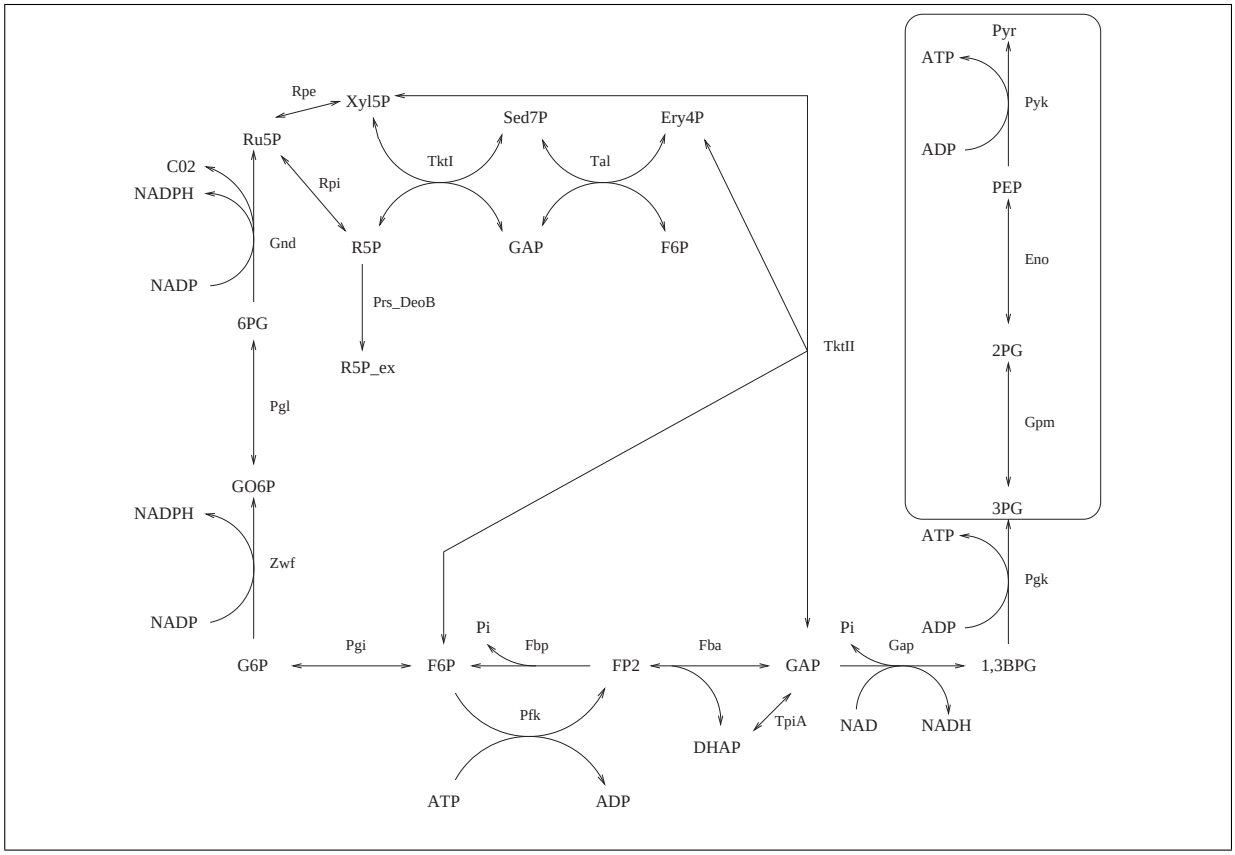


Fig. 5.3: Schéma réactionnel d'une partie du métabolisme des monosaccharides (pour des raisons de clarté, les composés *F6P* et *GAP* sont représentés 2 fois dans le schéma) (adapté de [Schuster *et al.*, 2000a])

La possibilité de prédire des chemins réactionnels non pertinents impose de développer des méthodes plus sophistiquées capables de prendre en compte ce type de réactions.

5.2 Approches contraintes par un équilibre global

Ces approches se basent sur le postulat suivant : "Une voie métabolique ne consomme et ne produit que les composés d'intérêt". En d'autres termes, étant donné un ensemble de composés d'intérêt, une voie métabolique intéressante ne produit et ne consomme que ces composés, en conséquence la consommation et la production des autres composés sont équilibrées (*"steady state"* en anglais). Respecter cette règle garantit que les solutions du problème, qui sont des réseaux réactionnels et non plus des chemins, seront acceptables d'un point de vue énergétique.

Certains métabolites jouent un rôle particulier et sont impliqués dans beaucoup de réactions. C'est le cas par exemple pour les coenzymes ou des petites molécules comme l'eau, le dioxyde de carbone ou encore l'ammoniac. Il est quasiment impossible de trouver

des réseaux métaboliques ne faisant pas apparaître ces composés dans leurs bilans. Ainsi, toutes les approches présentées ci-après doivent prendre en compte cette spécificité des réseaux métaboliques.

Ces approches utilisent comme point de départ un ensemble de réactions. Les métabolites sont considérés comme des ressources et les réactions sont des règles de production/consommation. A un ensemble de réactions données, il est ainsi possible d'associer une matrice "stœchiométrique".

Exemple : pour la partie encadrée du schéma de la figure 5.3, la matrice stœchiométrique décrivant le réseau est la suivante :

composés ↓	Pyk	Eno	Gpm	← réactions
Pyr	+1	0	0	
PEP	-1	+1	0	
2PG	0	-1	+1	
3PG	0	0	-1	
ATP	+1	0	0	
ADP	-1	0	0	

Dans cette matrice, à chaque composé correspond une ligne et à chaque réaction correspond une colonne. Un élément de la matrice a pour valeur le coefficient exprimant la consommation/production du métabolite concerné dans la réaction considérée. Il faut noter qu'aucune information sur la réversibilité des réactions n'est contenue dans cette matrice. Chacune des méthodes décrites ci-dessous possède sa propre manière de représenter les réactions réversibles et irréversibles.

Toutes ces méthodes nécessitent la définition de molécules "externes" et "internes". Les métabolites déclarés comme externes sont les entrées et les sorties du système tandis que les métabolites internes ne doivent pas intervenir dans le bilan global. Ces molécules internes doivent être soit inutilisées, soit produites et consommées dans les mêmes quantités.

[Fan *et al.*, 2002; Happel and Sellers, 1989; Küffner *et al.*, 2000; Mavrovouniotis, 1993; Schilling *et al.*, 2000; Schuster *et al.*, 2000a; Seressiotis and Bailey, 1988] posent le problème de la caractérisation d'un réseau métabolique en des termes très proches. Ils divergent néanmoins quelque peu dans leur formulation du problème : la table 5.1 donne un comparatif des différents travaux quant au problème exact qu'ils posent.

Référence	Spécification du problème	Condition sur l'ensemble solution
[Fan *et al.*, 2002]	– définition de la réaction finale recherchée	une solution est conservée si elle n'est pas une combinaison linéaire d'autres solutions
[Happel and Sellers, 1989]	– définition des composés internes et externes du système – ne tient pas compte de la réversibilité des réactions	une solution est conservée si elle n'est pas une combinaison linéaire d'autres solutions
[Küffner *et al.*, 2000]	– définition des métabolites internes et externes du système – définition, pour les composés externes, des substrats et des produits finaux des réseaux recherchés – définition de contraintes supplémentaires sur la topologie des solutions et notamment sur la taille (nombre de réactions) des solutions	l'ensemble des solutions est constitué de toutes les solutions du problème satisfaisant les contraintes
[Mavrovouniotis, 1993]	– définition des composés internes et externes du système – définition, pour les composés externes, des substrats et des produits finaux des réseaux recherchés – définition d'étiquettes de type *requis*, *autorisé* et *interdit* sur les métabolites et les réactions (les composés externes ne sont pas obligatoirement *requis*)	une solution est conservée si elle n'est pas une combinaison linéaire positive d'autres solutions
[Schilling *et al.*, 2000]	– définition des métabolites internes et externes du système	l'ensemble des solutions conservées sont les modes extrêmes du système (les vecteurs définissant les arêtes du cône des solutions du système)
[Schuster *et al.*, 2000a]	– définition des métabolites internes et externes du système	l'ensemble des solutions conservées sont les modes élémentaires (ensemble minimal de réactions pouvant fonctionner ensemble pour produire un état d'équilibre) du système
[Seressiotis and Bailey, 1988]	– définition des composés internes et externes du système – définition, pour les composés externes, des substrats et des produits finaux des réseaux recherchés	une solution est conservée si elle n'implique pas un ensemble de réactions plus grand qu'une autre solution

Tab. 5.1: Comparaison des travaux traitant du problème de la reconstruction contrainte par un équilibre global

Plusieurs auteurs ont noté le parallèle entre ces approches qui partagent de grandes similarités [Schuster *et al.*, 2002; 2000b]. Ce qui les différencie effectivement est l'objectif affiché (caractérisation ou reconstruction) et les algorithmes mis en œuvre (recherche exhaustive ou heuristique).

Suivant les travaux, trois types d'approches différentes sont utilisés pour résoudre le problème posé : l'approche algébrique, l'approche combinatoire et l'approche mixte.

5.2.1 Résolution algébrique

5.2.1.1 Réduction du problème de la reconstruction contrainte par un équilibre global au problème de la résolution d'un système d'inéquations linéaires

En utilisant la matrice stœchiométrique et des contraintes sur l'utilisation globale des composés, le problème de la reconstruction contrainte par un équilibre global peut se poser comme la résolution d'un système d'inéquations linéaires.

Tout d'abord chaque ligne définit une expression exprimant le nombre de molécules de chaque composé en fonction du nombre de fois où chaque réaction est utilisée.

Exemple : pour la matrice définie dans l'exemple précédent, les expressions définies sont les suivantes, où *#composé* représente le nombre de molécules produites/consommées du métabolites *composé* et *#réaction* le nombre de fois où la réaction *réaction* est utilisée :

$$\begin{array}{llll} \#Pyk & & & = \#Pyr \\ -\#Pyk & +\#Eno & & = \#PEP \\ & -\#Eno & +\#Gpm & = \#2PG \\ & & -\#Gpm & = \#3PG \\ \#Pyk & & & = \#ATP \\ -\#Pyk & & & = \#ADP \end{array}$$

Ensuite, il faut préciser les composés internes et externes. Cette information permet de définir les valeurs que doivent prendre les expressions définies précédemment.

Exemple : si les composés 2PG et PEP sont définis comme composés internes et les composés Pyr, 3PG, ATP et ADP comme externes, on obtient le système d'équations linéaires :

$$\begin{cases} -\#Pyk & +\#Eno & & = 0 \quad (PEP) \\ & -\#Eno & +\#Gpm & = 0 \quad (2PG) \end{cases}$$

En effet, pour PEP et 2PG, le fait d'avoir été désignés comme composés internes implique pour ces deux composés une consommation/production globale nulle.

Il faut également poser les contraintes qui garantissent que les réactions irréversibles ne sont pas utilisées dans le mauvais sens. Pour l'exemple choisi, cela se traduit par l'ajout d'une inéquation $\#Pyk \geq 0$ car la réaction Pyk est irréversible. Il est également possible d'enrichir ce système par des contraintes supplémentaires. Par exemple, il est possible d'imposer que le composé Pyr doit être globalement produit tandis que le composé 3PG doit être globalement consommé. Ces deux contraintes sont traduites par deux inéquations. Cela donne comme système d'inéquations linéaires final le système suivant :

$$\begin{cases} -\#Pyk \quad +\#Eno & & = 0 \quad (PEP) \quad (1) \\ \quad\quad -\#Eno & +\#Gpm & = 0 \quad (2PG) \quad (1) \\ \\ \#Pyk & & \geq 0 \quad (Pyk) \quad (2) \\ \\ \#Pyk & & > 0 \quad (Pyr) \quad (3) \\ & -\#Gpm & < 0 \quad (3PG) \quad (3) \end{cases}$$

Le système d'inéquations linéaires est donc défini à partir de la matrice stœchiométrique et des contraintes sur les types des composés (1), des contraintes sur la réversibilité des réactions (2) et des contraintes sur la consommation globale de certains composés externes (3).

La résolution du système d'inéquations linéaires ainsi défini n'a pas forcément de solution. Chaque solution est un vecteur de réels qui correspond à l'utilisation de chaque réaction dans un état d'équilibre. Un élément nul dans ce vecteur indique que la réaction associée est inutilisée par le réseau réactionnel solution correspondant.

Le problème générique à résoudre est le suivant :

PROBLÈME **1** RÉSOLUTION DE SYSTÈMES D'INÉQUATIONS LINÉAIRES
DONNÉES : *un système d'inéquations linéaires avec les relations* $= 0$, > 0, < 0, ≥ 0 *et* ≤ 0 *défini sur les* n *variables* $N = \{n_1, \ldots, n_n\}$
RÉPONSE : *le sous-ensemble des solutions* $\overrightarrow{V} \in \mathbb{R}^{|N|}$ *caractéristique de toutes les solutions du système*

Un vecteur solution du système précédent est toujours défini à une constante multiplicative près (si $\overrightarrow{V}$ est solution alors $\alpha.\overrightarrow{V}$, $\alpha \neq 0$ est également solution). On peut donc se limiter à un ensemble de vecteurs caractéristiques de l'ensemble des vecteurs solutions du système. L'espace des solutions des systèmes construits à partir des contraintes énoncées ci-dessus peut être représenté par une pyramide à base convexe (appelée cône des solutions) (voir figure 5.4). Il existe de nombreux algorithmes pour calculer les vecteurs qui définissent les arêtes de ce cône [Clarke, 1988]. Comme le cône contient l'ensemble de toutes les solutions, n'importe quelle combinaison linéaire à coefficients positifs des vecteurs définissant le cône reste à l'intérieur du cône et est solution du système.

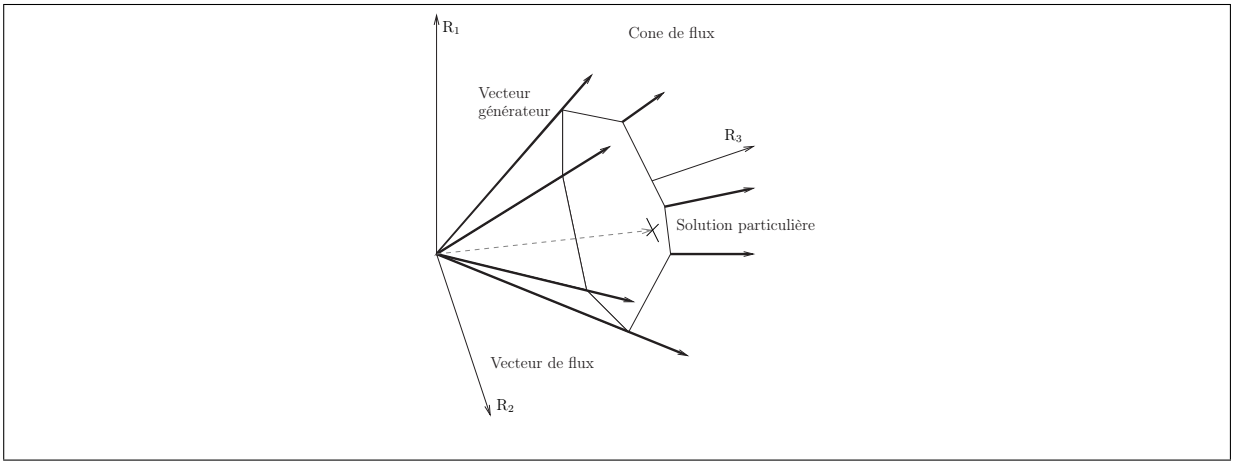


Fig. 5.4: Représentation du cône des solutions d'un système d'inéquations linéaires construit pour la caractérisation d'un réseau métabolique (adapté de [Schilling *et al.*, 2000])

La plupart des approches qui se basent sur cette modélisation du problème n'ont pas pour objectif de résoudre le problème de la reconstruction. Elles ont pour but de caractériser un réseau métabolique précis, c'est-à-dire qu'étant donné un réseau métabolique (décrit par les réactions impliquées), on veut obtenir toutes les modes de fonctionnement de ce réseau (toutes les possibilités de fabrication des métabolites sortants par les métabolites entrants). Dans ce cas, le réseau est décrit par un ensemble réduit de réactions.

5.2.1.2 Les méthodes de résolution

[Happel and Sellers, 1989; Schilling *et al.*, 2000; Schuster *et al.*, 2000a] proposent une approche algébrique pour résoudre le problème de la reconstruction, c'est-à-dire, qu'ils ne manipulent qu'une représentation matricielle des données par des opérations de combinaisons entre colonnes ou entre lignes. Leur but est de diagonaliser ou de rendre nul tous les coefficients d'un sous-tableau d'un tableau initial construit à partir de la matrice stœchiométrique.

Le travail le moins adapté à la recherche de voies métaboliques est [Happel and Sellers, 1989] car il ne prend pas en compte la réversibilité des réactions. Les approches [Schilling *et al.*, 2000] et [Schuster *et al.*, 2000a] sont très voisines et [Schilling *et al.*, 1999] montrent quels sont les points communs et les divergences de ces deux approches. [Schilling *et al.*, 2000] donne comme résultat l'ensemble des vecteurs représentant les arêtes du cône de flux représentant l'espace des solutions. [Schuster *et al.*, 2000a] donne comme résultat les *modes élémentaires* du système.

Définition **5** *Mode élémentaire*
Le vecteur V solution du système est un mode élémentaire si et seulement si il correspond

à une distribution de flux (les arêtes du cône des solutions) minimale en terme de réactions utilisées

Note : toutes les arêtes du cône des solutions sont des modes élémentaires

La différence majeure des deux approches est que l'ensemble résultat fourni par [Schilling *et al.*, 2000] est toujours inclus dans (ou égal à) celui fourni par [Schuster *et al.*, 2000a]. La figure 5.5 montre un cas où le résultat fourni par les deux algorithmes de résolution décrits dans [Schilling *et al.*, 2000] et [Schuster *et al.*, 2000a] n'est pas le même. Les deux flux en pointillés courts forment la base convexe de l'ensemble des solutions (résultat donné par [Schilling *et al.*, 2000]). Pour les modes élémentaires (résultat donné par [Schuster *et al.*, 2000a]), il y a un flux supplémentaire, représenté en pointillés longs.

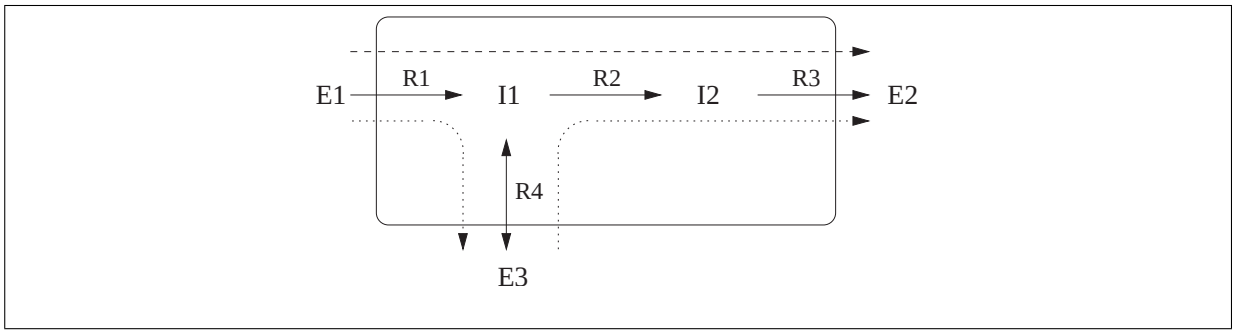


Fig. 5.5: Un système de réaction simple qui montre la différence entre les modes extrêmes et les modes élémentaires - La réaction avec la flèche double est considérée comme réversible, les E_i sont les composés externes du système, les I_i les composés internes. Dans ce cas, il y a trois modes élémentaires alors que le cône solution n'a que deux dimensions (adapté de [Schuster *et al.*, 2002])

L'algorithme utilisé dans [Schuster *et al.*, 2000a] est décrit en détail, cet algorithme a été implémenté dans [Pfeiffer *et al.*, 1999]. Cet algorithme calcule tous les modes élémentaires d'un réseau de réactions. Pour illustrer le déroulement de l'algorithme, l'ensemble de réactions du réseau de la figure 5.3 servira de référence. En groupant les réactions qui doivent forcément fonctionner ensemble, on obtient le réseau réduit de la figure 5.6 qui contient 9 réactions. Dans cette figure, les composés définis comme externes sont indiqués en gras.

5.2.1.3 Initialisation

A partir de la liste des réactions et de l'information concernant leur réversibilité ainsi que la liste des composés internes et externes, un tableau initial $T^{(0)}$ est constitué. Il est composé de la concaténation de la transposée de la matrice stœchiométrique, sans les colonnes correspondant aux composés externes, et de la matrice identité. Les colonnes les

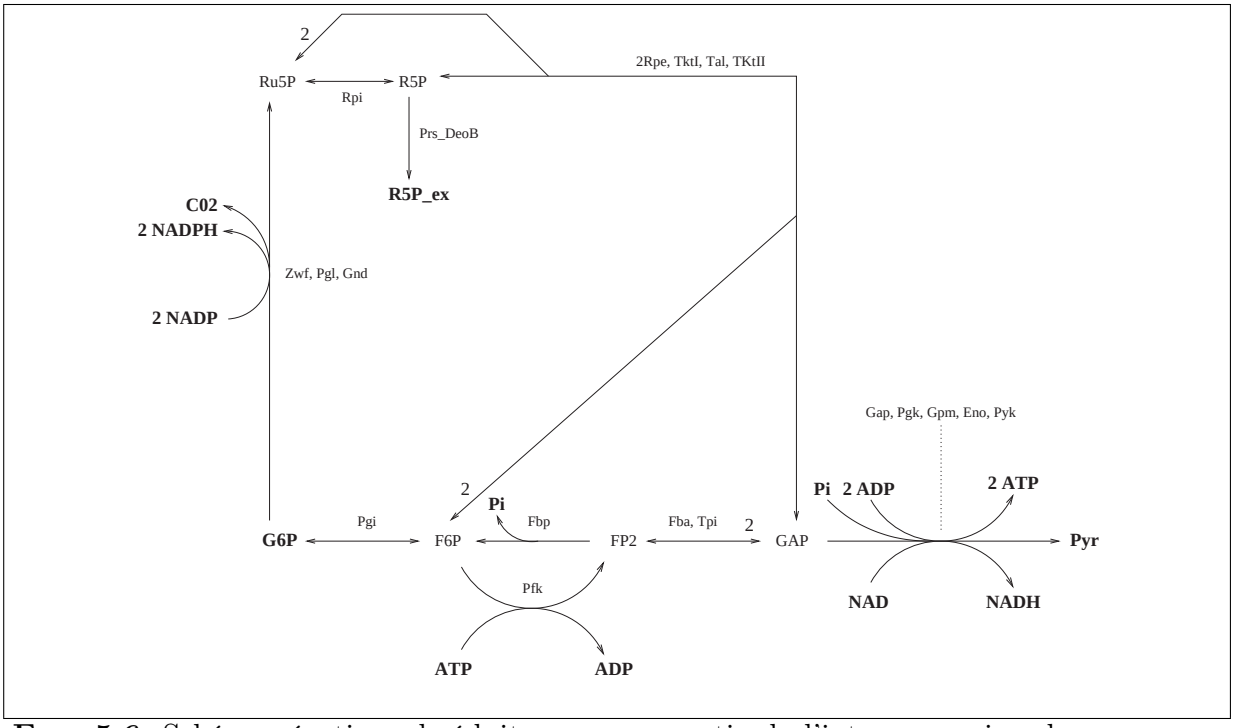


FIG. 5.6: Schéma réactionnel réduit pour une partie de l'interconversion des monosaccharides (adapté de [Schuster *et al.*, 2000a])

plus à gauche du tableau correspondent aux composés internes, celles de droite forment la matrice identité. Le tableau est décomposé en deux parties horizontales, une concerne les réactions réversibles, l'autre les réactions irréversibles. Le tableau initial correspondant au schéma de la figure 5.6 est le suivant (adapté de [Schuster *et al.*, 2000a]).

$$T^{(0)} = \overbrace{\begin{array}{ccccc} \text{Ru5P} & \text{FP2} & \text{F6P} & \text{GAP} & \text{R5P} \end{array}}^{\text{composés internes}}$$

$$\left[\begin{array}{ccccc|cccccccccc} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 2 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$\begin{array}{l} \leftarrow Pgi \\ \leftarrow \{Fba, TpiA\} \\ \leftarrow Rpi \\ \leftarrow \{2Rpe, TklI, Tal, TklII\} \\ \hline \leftarrow \{Gap, Pgk, Gpm, Eno, Pyk\} \\ \leftarrow \{Zwf, Pgl, Gmd\} \\ \leftarrow Pfk \\ \leftarrow Fbp \\ \leftarrow Prs\_DeoB \end{array}$$

réactions réversibles

réactions irréversibles

La concaténation avec la matrice identité permet d'associer à chaque réaction une colonne qui permet de l'identifier. Cette partie du tableau permet, à chaque étape de l'algorithme, de connaître l'implication de chaque réaction dans chaque ligne du tableau. A la fin de l'algorithme, on connaît alors pour chaque ligne restante (qui correspond à un mode élémentaire) l'implication de chaque réaction.

5.2.1.4 Schéma d'algorithme

Par définition, un mode élémentaire ne fait intervenir aucun composé interne dans son bilan. Le principe de l'algorithme est de calculer itérativement un tableau contenant les réactions et les combinaisons de réactions qui ne font pas intervenir un composé interne choisi. A chaque itération, le nouveau tableau est calculé à partir de celui calculé à l'itération précédente. A la i$^{\text{ème}}$ étape de l'algorithme, on calcule donc les combinaisons de réactions qui ne font intervenir ni le premier composé interne, ni le deuxième ... jusqu'au i$^{\text{ème}}$ composé interne. Dans le tableau final, chaque ligne représente les combinaisons de réactions qui ne font intervenir aucun composé interne dans leur bilan, il s'agit des modes élémentaires.

Chaque nouveau tableau a donc une colonne à gauche nulle supplémentaire, car chaque colonne à gauche correspond à l'utilisation d'un composé interne dans les bilans des combinaisons de réactions. A la fin, le tableau final aura la sous matrice gauche nulle. Toutes les combinaisons de réactions calculées restantes seront les modes élémentaires. Le tableau $T^{(i)}$ calculé à l'étape i est obtenu en combinant des lignes, qui correspondent à une utilisation de réactions, du tableau $T^{(i-1)}$ calculé à l'étape $i-1$. La sous-matrice droite du tableau (initialement la matrice identité) garde la trace des combinaisons qui ont été faites. Toutes les combinaisons de lignes possibles sont calculées mais chaque combinaison doit vérifier trois conditions :

1. les lignes correspondant à des réactions irréversibles ne peuvent être qu'ajoutées et non retranchées dans les combinaisons
2. une combinaison de deux lignes de $T^{(i-1)}$ est ajoutée à $T^{(i)}$ seulement si la ligne résultat n'implique pas un ensemble de réactions plus grand que celui d'une ligne déjà existante de $T^{(i)}$
3. une combinaison de deux lignes de $T^{(i-1)}$ ajoutée à $T^{(i)}$ peut devenir incorrecte si, pour le calcul du tableau $T^{(i)}$, on ajoute une ligne qui implique un ensemble de réactions plus petit que celui impliqué par cette combinaison. Dans ce cas, il faut enlever la ligne précédemment insérée

5.2.1.5 Exemple d'exécution

Pour calculer les modes élémentaires du réseau de la figure 5.6, on commence par construire le tableau initial $T^{(0)}$. A partir du tableau $T^{(0)}$, on peut calculer le tableau $T^{(1)}$, dont la première colonne est nulle, qui vaut[1] :

[1] on note $T(i,.)$ la i$^{\text{ème}}$ ligne du tableau T

$$T^{(1)} = \left[\begin{array}{ccccc|cccccccccc}
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -2 & -1 & +3 & 0 & 0 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 2 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 0 & 0
\end{array}\right]$$

$\leftarrow T^{(0)}(1,.)$
$\leftarrow T^{(0)}(2,.)$
$\leftarrow 2 \times T^{(0)}(3,.) + (-1) \times T^{(0)}(4,.)$

} combinaisons réversibles

$\leftarrow T^{(0)}(5,.)$
$\leftarrow T^{(0)}(7,.)$
$\leftarrow T^{(0)}(8,.)$
$\leftarrow T^{(0)}(9,.)$
$\leftarrow T^{(0)}(3,.) + T^{(0)}(6,.)$
$\leftarrow T^{(0)}(4,.) + 2 \times T^{(0)}(6,.)$

} combinaisons irréversibles

Les lignes qui correspondent à des combinaisons sont les lignes :

- $T^{(1)}(3,.) = 2 \times T^{(0)}(3,.) + (-1) \times T^{(0)}(4,.)$
- $T^{(1)}(8,.) = T^{(0)}(3,.) + T^{(0)}(6,.)$
- $T^{(1)}(9,.) = T^{(0)}(4,.) + 2 \times T^{(0)}(6,.)$

Les autres lignes sont issues du tableau $T^{(0)}$ car les réactions correspondantes ne font pas intervenir le composé Ru5P dans leur bilan.

Le tableau $T^{(2)}$, dont les deux premières colonnes sont nulles contient les lignes $T^{(1)}(1,.)$, $T^{(1)}(3,.)$, $T^{(1)}(4,.)$, $T^{(1)}(7,.)$, $T^{(1)}(8,.)$ et $T^{(1)}(9,.)$ car ces lignes ont déjà des 0 dans leur deuxième colonne. Le tableau $T^{(2)}$ vaut :

$$T^{(2)} = \left[\begin{array}{ccccc|cccccccccc}
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -2 & -1 & +3 & 0 & 0 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 2 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & -1 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & -2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{array}\right]$$

Finalement, on obtient au dernier tableau $T^{(5)}$ dont les lignes sont les modes élémentaires.

$$T^{(5)} = \left[\begin{array}{ccccc|cccccccccc}
\hline
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 2 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -2 & 0 & 1 & 1 & 1 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 5 & 3 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 5 & 1 & 4 & -2 & 0 & 0 & 1 & 0 & 6 \\
0 & 0 & 0 & 0 & 0 & -5 & -1 & 2 & 2 & 0 & 6 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{array}\right]$$

Dans le tableau final, on remarque que tous les modes élémentaires calculés sont irréversibles et qu'il y en a sept. Ces sept modes élémentaires correspondent aux sous-réseaux de la figure 5.7.

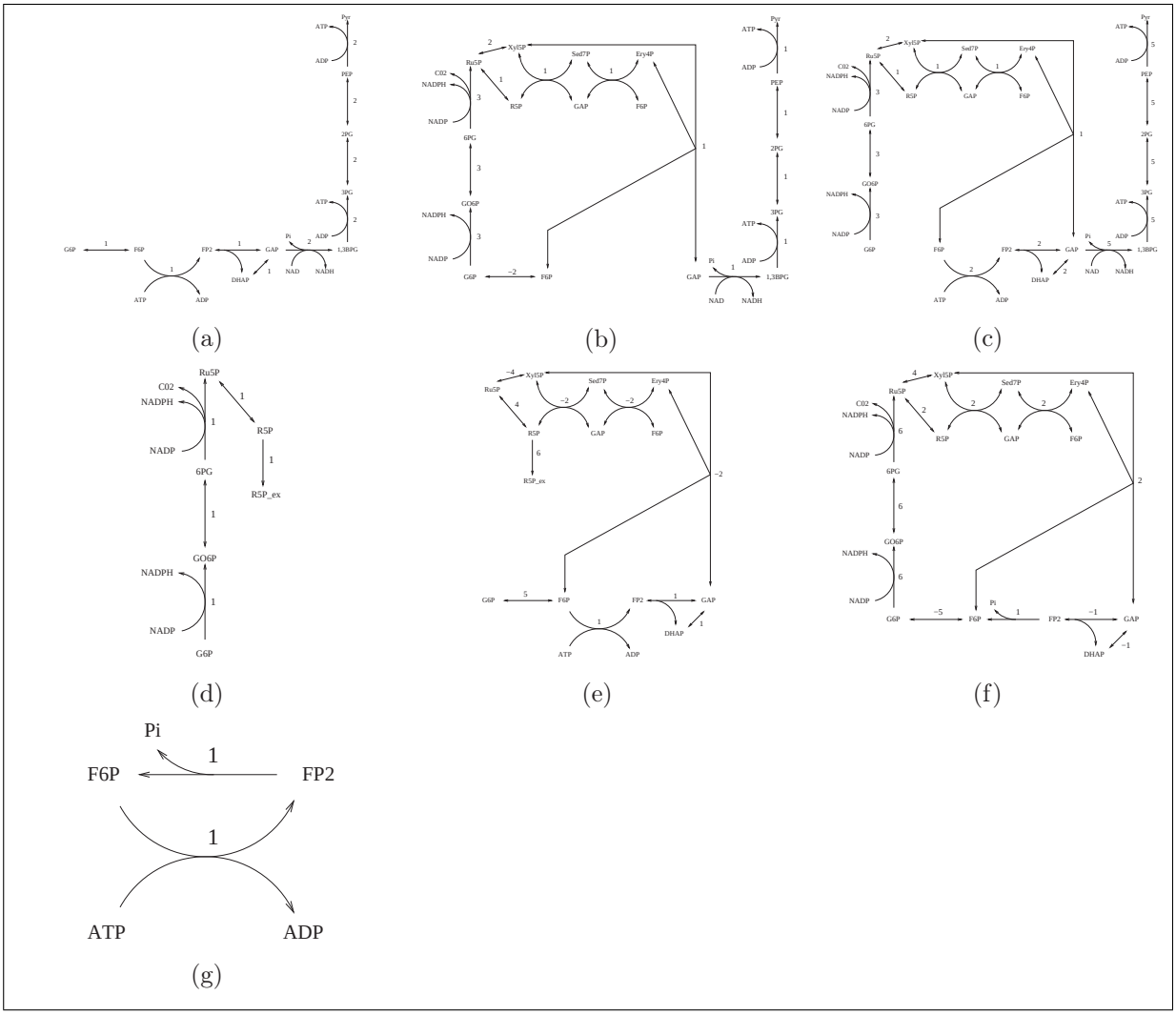


Fig. 5.7: Représentation des sept modes élémentaires du réseau de la figure 5.3 - Le coefficient qui est associé à chaque réaction est inscrit à la place des noms des enzymes (adapté de [Schuster *et al.*, 2000a])

5.2.2 Résolution combinatoire

[Küffner *et al.*, 2000; Mavrovouniotis, 1993] utilisent une approche combinatoire pour résoudre un problème proche du problème 1 défini au paragraphe précédent. La méthode de résolution utilisée dans [Küffner *et al.*, 2000] consiste à énumérer tous les réseaux possibles qui satisfont certaines contraintes supplémentaires posées (voir tableau 5.1). Cela revient à une résolution incomplète du problème de base car les contraintes supplémentaires réduisent l'espace des solutions. Dans leur travail, l'accent porte sur la possibilité de prendre en compte un grand nombre de réactions (plusieurs milliers). Le but de ce travail n'est donc pas de résoudre exactement (c'est-à-dire complètement) le problème 1, mais plutôt de le résoudre partiellement en fournissant un ensemble de solutions respectant

un ensemble de contraintes d'ordre topologique, les caractéristiques de ces solutions les rendant de fait intéressantes. Ce travail est décrit dans le § suivant.

5.2.2.1 Recherche de sous-réseaux contraints

[Küffner *et al.*, 2000] utilise les réseaux de Petri (voir § 3.2.1)comme modèle pour manipuler les réseaux métaboliques (voir § 3.2.2). La formulation du problème résolu dans [Küffner *et al.*, 2000] se base sur les définitions suivantes[2] :

DÉFINITION **6** *Chemin dans un réseau de Petri*
Dans un réseau de Petri $R = (P, T, Pre, Post)$, un chemin entre deux places $A \in P$ et $B \in P$ est une succession de transitions $[t_1, \ldots, t_n]$ telle que :

- *à une place sortante de t_i correspond une place entrante de t_{i+1}*
- *A est une place entrante de t_1*
- *B est une place sortante de t_n*

DÉFINITION **7** *Vecteur de transitions clos*
Etant donnés un réseau de Petri $R = (P, T, Pre, Post)$, un vecteur V d'entiers positifs ou nuls de taille $|T|$ est dit clos par rapport à un ensemble de places $Q \subseteq P$ si et seulement si :

$$C(p, .) \cdot V = 0 \qquad \forall p \in P \setminus Q$$

où C est la matrice d'incidence de R, $C(p, .)$ est le vecteur ligne d'indice p de C

Exemple : la figure 5.8 illustre cette définition. Dans cet exemple, les vecteurs V_1 et V_2 définissent des utilisations des transitions du réseau de Petri qui garantissent que toutes les ressources associées aux places du réseau (sauf les grises) sont autant produites que consommées.

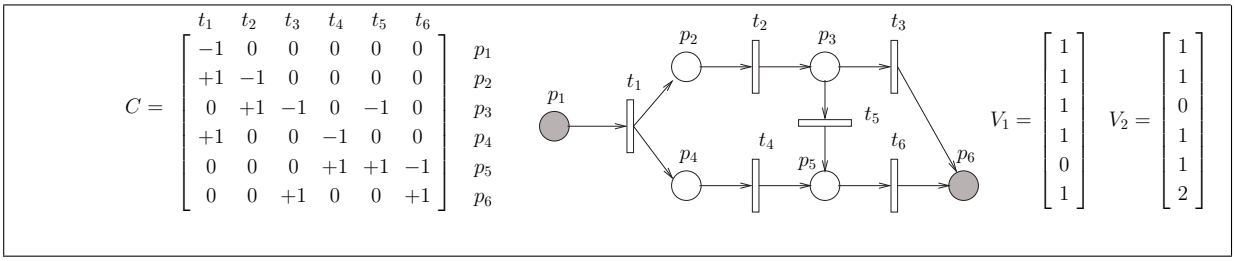


FIG. 5.8: Exemple de vecteurs clos par rapport à l'ensemble des places grises pour le réseau de Petri décrit par la matrice C

[2]Dans l'article original, les définitions et l'algorithme mis en œuvre ne coïncident pas. Les définitions présentées ici ont donc été adaptées pour que le problème posé soit bien résolu par l'algorithme présenté.

Les vecteurs V_1 et V_2 de la figure 5.8 sont clos pour le réseau de Petri décrit par la matrice C par rapport aux places $\{p_1, p_6\}$ car $C \times V_1 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \end{bmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{matrix}$ et $C \times V_2 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \end{bmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{matrix}$.

DÉFINITION **8** *Vecteur de transitions clos minimal*
Etant donnés un réseau de Petri $R = (P, T, Pre, Post)$, un vecteur V d'entiers positifs ou nuls de taille $|T|$ est dit clos minimal par rapport à un ensemble de places $Q \subseteq P$ si et seulement si :

- *V est clos par rapport à Q*
- *il n'existe aucun vecteur V' clos par rapport à Q tel que $V' \leq V$ ($V' \leq V$ ssi $\forall i, V'[i] \leq V[i]$)*

Exemple : la figure 5.9 illustre cette définition.

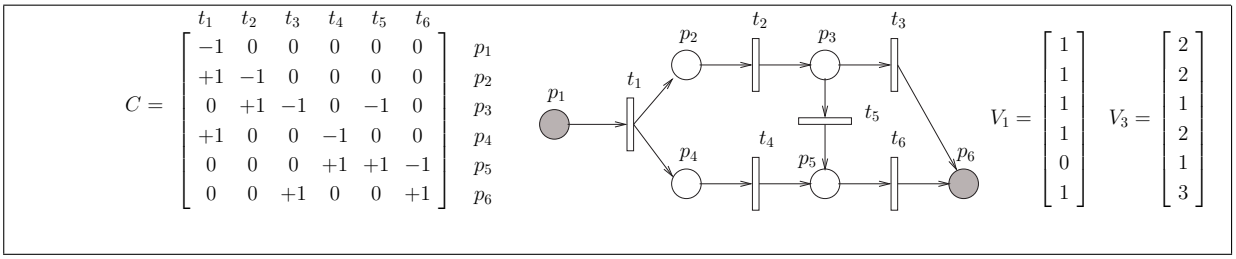


FIG. 5.9: Exemple de vecteurs clos minimaux et non minimaux par rapport à l'ensemble des places grises pour le réseau de Petri décrit par la matrice C

Les vecteurs V_1 et V_3 de la figure 5.9 sont clos par rapport aux places $\{p1, p6\}$ mais V_3 n'est pas minimal car $V_1 \leq V_3$.

DÉFINITION **9** *Sous-réseau induit par un vecteur*
Etant donnés un réseau de Petri $R = (P, T, Pre, Post)$ et un vecteur V d'entiers positifs ou nuls de taille $|T|$. On définit l'ensemble T' de transitions comme suit :

$$\begin{aligned} V[t] &\neq 0 \text{ si } t \in T' \\ &= 0 \text{ sinon} \end{aligned}$$

Le réseau induit par T' est appelé sous-réseau induit par V.

Ces définitions permettent de poser le problème générique résolu dans [Küffner *et al.*, 2000].

PROBLÈME **2** RECHERCHE DE VECTEURS CLOS MINIMAUX DANS UN RÉSEAU DE PETRI
DONNÉES : *un réseau de Petri $R = (P, T, Pre, Post)$, deux places $A \in P$ et $B \in P$, un*

ensemble de places $U \subseteq P \setminus \{A, B\}$
RÉPONSE : *l'ensemble des vecteurs* V *tel que* V *est clos par rapport à* $U \bigcup \{A, B\}$ *et tel que chaque sous-réseau induit par* V *contient un chemin de* A *à* B

Le problème de base est enrichi par des contraintes supplémentaires illustrées sur la figure 5.10.

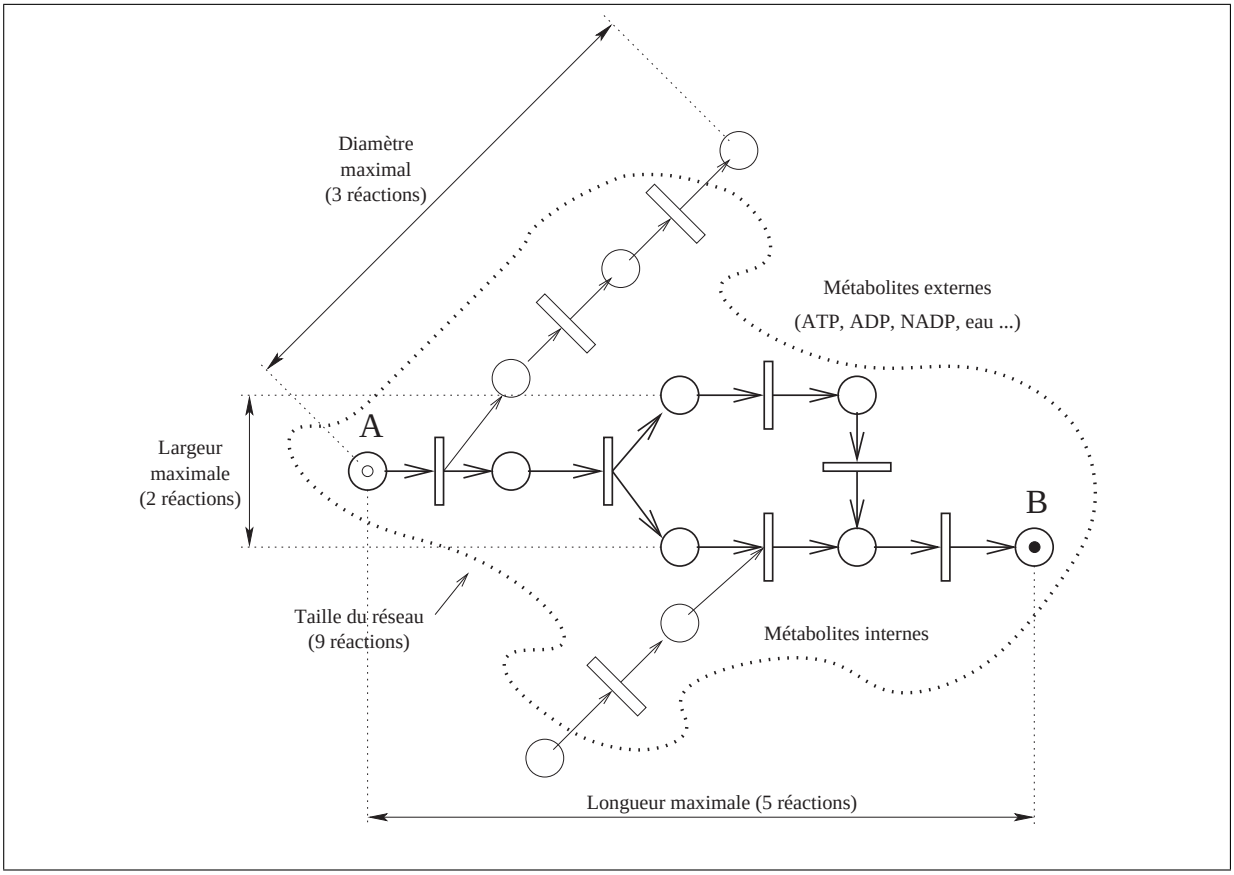


FIG. 5.10: Ensemble des contraintes utilisées dans [Küffner *et al.*, 2000] - Indications des mesures pour l'exemple choisi (adapté de [Küffner *et al.*, 2000])

Le problème exactement posé et résolu est donc le suivant :

PROBLÈME **3** RECHERCHE DE VECTEURS CLOS MINIMAUX ET CONTRAINTS DANS UN RÉSEAU DE PETRI
DONNÉES : *un réseau de Petri* $R = (P, T, Pre, Post)$*, deux places* $A \in P$ *et* $B \in P$*, un ensemble de places* $U \subseteq P \setminus \{A, B\}$ *et des entiers strictement positifs* l, a, d, w
RÉPONSE : *l'ensemble des vecteurs* V *tel que* V *est clos minimal par rapport à* $U \bigcup \{A, B\}$ *et tel que chaque sous-réseau* $R' = (P', T', Pre', Post')$ *induit par* V *satisfait les contraintes :*

- R' *contient un chemin de* A *à* B
- *le plus long des chemins de* A *à* B *a une longueur inférieure ou égale à* l

- *R' a un nombre total de transitions inférieur ou égal à a*
- *le diamètre du réseau (longueur maximale des chemins entre A et les composés en excès et entre B et les composés en excès) doit être inférieur ou égal à d*
- *la largeur du réseau (coupe de taille maximale entre A et B) doit être inférieure ou égale à w*

De plus, comme dans [Mavrovouniotis, 1993], il est permis de définir des étiquettes, à la fois sur les composés et les réactions. Ces étiquettes peuvent être de deux types :

- exclusion
- inclusion

Elles spécifient l'obligation que les sous-réseaux induits excluent ou intègrent les éléments étiquetés.

5.2.2.2 Algorithme

L'algorithme utilisé dans [Küffner *et al.*, 2000] pour résoudre le problème 3 est composé de deux parties. La première réduit la taille du réseau initial pour qu'il ne contienne que les transitions qui sont susceptibles de faire partie d'une solution. Cette restriction utilise les contraintes imposées sur la longueur maximale du chemin entre les deux extrémités de la voie et le diamètre de la voie. La seconde partie énumère par l'intermédiaire d'un algorithme de type Branch&Bound les solutions dans le réseau calculé à l'étape précédente.

Réduction du réseau initial La prise en compte des contraintes sur la longueur et le diamètre des réseaux solutions, illustrées par la figure 5.10, ainsi que les étiquettes, permettent de réduire le réseau initial dans lequel les sous réseaux sont recherchés.

La procédure est la suivante :

1. toutes les places et les transitions étiquetées comme *exclues* sont enlevées
2. en partant de la source A, déterminer pour chaque transition une borne inférieure sur le nombre de transitions à tirer pour l'atteindre
3. déterminer pour chaque transition une borne inférieure sur le nombre de transitions à tirer pour atteindre B
4. en partant de la source A ou des composés de U, déterminer pour chaque transition une borne inférieure sur le nombre de transitions à tirer pour l'atteindre
5. déterminer pour chaque transition une borne inférieure sur le nombre de transitions à tirer pour atteindre B ou un des éléments de U
6. sur la base des valeurs calculées aux étapes 2 3 4 et 5, une transition est enlevée si elle ne peut pas être sur un chemin de longueur au plus l de A à B et si elle ne peut

pas non plus être sur un chemin de A à un des éléments de U ou d'un des éléments de U à B dans les limites du diamètre autorisé. Tant qu'au moins une transition est supprimée, recommencer à l'étape 2

Algorithme Les solutions recherchées par l'algorithme sont les vecteurs clos minimaux du réseau de Petri $R = (P, T, Pre, Post)$ qui satisfont l'ensemble des contraintes supplémentaires. L'algorithme manipule donc un ensemble de vecteurs de taille $|T|$ d'entiers positifs ou nuls qui représentent chacun une utilisation différente des transitions du réseau R. A chaque étape, un vecteur est choisi et enlevé des vecteurs en cours de traitement. Il est modifié pour rendre compte de l'utilisation supplémentaire d'une transition et comme tous les cas possibles sont explorés, pour un vecteur choisi, plusieurs nouveaux vecteurs sont créés. Si un nouveau vecteur est solution du problème, il est conservé à part, sinon, il est inséré dans l'ensemble des vecteurs en cours de traitement. Le processus est initialisé avec un ensemble contenant seulement le vecteur nul.

5.2.3 Résolution mixte

[Fan *et al.*, 2002] n'est pas un travail à proprement parler dédié à l'étude de systèmes de réactions enzymatiques même s'il a déjà été appliqué à des ensembles de réactions biochimiques [Seo *et al.*, 2001]. Ce travail visait initialement l'étude de systèmes de synthèse de composés chimiques à partir d'autres composés chimiques. En général, de telles études ne font pas intervenir un très grand nombre de réactions. Par exemple, pour l'étude de la synthèse de l'ammoniac, le système considéré dans [Fan *et al.*, 2002] ne contient que 14 réactions élémentaires, toutes réversibles. Comme elle a été conçue pour étudier des systèmes chimiques dont le bilan réactionnel est assez simple, cette approche réclame le bilan exact du fonctionnement du réseau recherché.

Exemple : dans le cas de la synthèse de l'ammoniac l'équation bilan est $N_2 + 3H_2 \leftrightarrows 2NH_3$.

Une particularité de cette approche est que la résolution du problème se décompose en plusieurs étapes successives qui font intervenir deux types différents de méthode de résolution. Les premières étapes font intervenir des algorithmes de résolution combinatoire de type Branch&Bound, alors que la dernière étape se base sur la résolution de systèmes d'équations linéaires.

[Fan *et al.*, 2002] se base sur une représentation des réseaux réactionnels appelés P-graphes. Avant de s'intéresser à l'enchaînement et à la description des différentes étapes de la résolution du problème, nous allons présenter ces P-graphes.

5.2.3.1 Définition des P-graphes

P-graphe est la contraction de *process graph*. Ces graphes ont été à l'origine introduits pour décrire les processus de synthèse de produits chimiques. Les P-graphes sont très similaires aux réseaux de Petri présentés au § 5.2.2. Ce sont des graphes bipartites. Un type de nœuds est réservé aux espèces chimiques et un autre aux réactions. La différence entre un P-graphe et un réseau de Petri est que dans un P-graphe, les transitions ne sont pas valuées.

DÉFINITION **10** *P-graphe*
Un P-graphe est un doublet (M, O) *où :*
- $M = \{m_1, \ldots, m_n\}$
- $O = \{(p_1, p_2), \ldots, (p_{r-1}, p_r)\}$ *où* $\forall i, p_i \subseteq M$

M est l'ensemble des nœuds représentant les espèces chimiques
O est l'ensemble des nœuds représentant les réactions chimiques

5.2.3.2 Représentation graphique des P-graphes

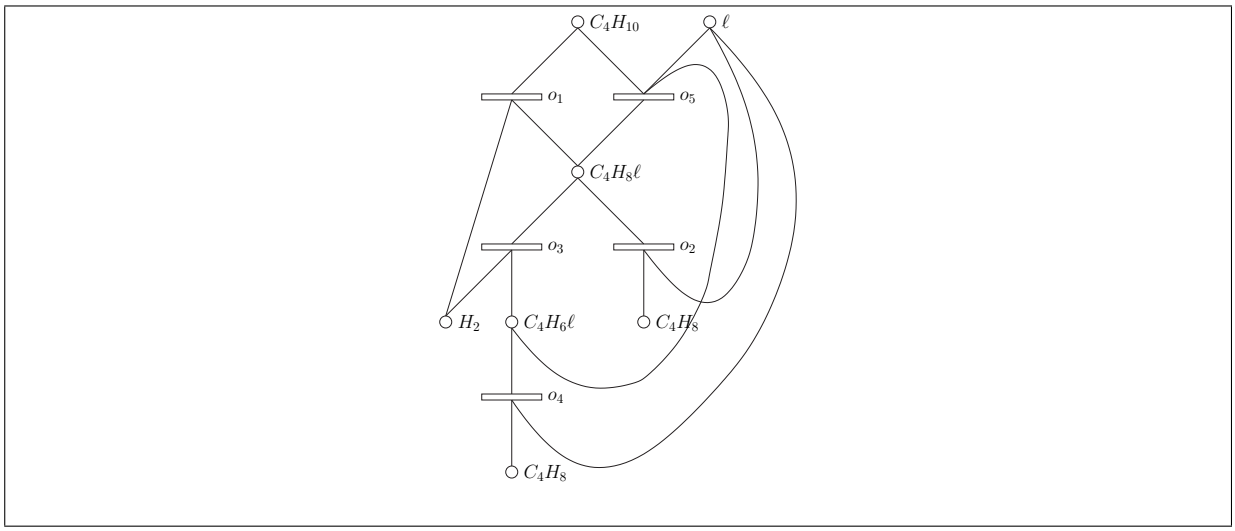


FIG. 5.11: Représentation graphique d'un P-graphe (adapté de [Fan *et al.*, 2002]) - Aucune transition n'est explicitement orientée, néanmoins, les espèces chimiques intervenant dans une réaction sont séparées suivant que le liens entre la réaction et les composés se branchent sous ou au dessus du rectangle représentant la réaction

Les P-graphes se représentent de la même façon que les réseaux de Petri.

Exemple : la figure 5.11 montre la représentation graphique du P-graphe
$P = (M = \{C_4H_{10}, C_4H_8\ell, C_4H_8, C_4H_6\ell, C_4H_6, H_2, \ell\}, O = \{o_1, o_2, o_3, o_4, o_5\})$
où

- $o_1 = (\{C_4H_{10}, \ell\}, \{C_4H_8\ell, H_2\})$
- $o_2 = (\{C_4H_8\ell\}, \{C_4H_8, \ell\})$
- $o_3 = (\{C_4H_8\ell\}, \{C_4H_6\ell, H_2\})$
- $o_4 = (\{C_4H_6\ell\}, \{C_4H_6, \ell\})$
- $o_5 = (\{C_4H_{10}, \ell, C_4H_6\ell\}, \{C_4H_8\ell\})$

les o_i sont associés aux réactions suivantes :

- $r_1 = C_4H_{10} + \ell \leftrightarrows C_4H_8\ell + H_2$
- $r_2 = C_4H_8\ell \leftrightarrows C_4H_8 + \ell$
- $r_3 = C_4H_8\ell \leftrightarrows C_4H_6\ell + H_2$
- $r_4 = C_4H_6\ell \leftrightarrows C_4H_6 + \ell$
- $r_5 = C_4H_{10} + \ell + C_4H_6\ell \leftrightarrows 2C_4H_8\ell$

ces réactions sont impliquées dans la déshydrogénation du butane en butène en présence d'un catalyseur ℓ. Ce problème est également étudié avec l'approche décrite dans [Fan *et al.*, 2002].

Cela permet de poser le problème résolu dans [Fan *et al.*, 2002] :

Problème **4** Système de synthèse de composés chimiques
Données : *un ensemble R de réactions (toutes considérées comme réversibles) faisant intervenir l'ensemble des composés $C = \{c_1, \ldots, c_{|C|}\}$ et une équation bilan Q (de la forme $n_ic_i + \ldots + n_jc_j \leftrightarrows n_kc_k + \ldots + n_lc_l$)*
Réponse : *l'ensemble des utilisations des réactions de R (un vecteur de taille $|R|$ composés d'entiers) qui ont pour bilan l'équation bilan Q.*

5.2.3.3 Présentation générale de l'algorithme

L'algorithme fonctionne en trois étapes successives :

1. la première étape (réduction du réseau) consiste, à partir d'un P-graphe construit à partir d'un ensemble de réactions (toutes réversibles) et d'un bilan à atteindre, à supprimer des réactions qui ne peuvent pas faire partie de la solution du problème
2. la deuxième étape (construction des réseaux candidats) consiste, à partir d'un P-graphe et d'un bilan à atteindre, à rechercher des sous-réseaux candidats du P-graphe initial
3. la dernière étape (évaluation des réseaux candidats) consiste à valider ou à rejeter les sous-réseaux candidats proposés en établissant le nombre de fois où chaque réaction doit être utilisée pour atteindre le bilan fixé. C'est cette dernière étape qui utilise la résolution de systèmes linéaires

5.2.3.4 Formulation axiomatique du problème

Dans [Fan *et al.*, 2002], le problème est caractérisé par un ensemble de propriétés que les solutions doivent forcément vérifier. Elles sont au nombre de cinq :

1. chaque produit final est totalement produit par les réactions de la solution
2. chaque substrat initial est totalement consommé par les réactions de la solution
3. chaque composé chimique produit par une réaction de la solution doit être totalement consommé par une ou plusieurs réactions du réseau solution. Chaque composé chimique consommé par une réaction de la solution doit être totalement produit par une ou plusieurs réactions du réseau solution
4. le réseau représentant l'ensemble des réactions de la solution est acyclique
5. au moins une réaction représentée dans le réseau solution consomme un substrat initial

Les quatre premières règles peuvent être utilisées afin de donner l'ensemble des conditions suivantes que doivent vérifier les sous-réseaux candidats générés lors de la deuxième étape de la résolution du problème :

1. chaque produit final est présent dans le réseau solution
2. chaque substrat initial est présent dans le réseau solution
3. chaque composé chimique présent dans le réseau possède un chemin menant à un produit final du problème
4. un composé chimique représenté dans le réseau solution est un substrat initial s'il n'est produit par aucune réaction du réseau
5. le réseau solution n'implique une réaction que dans un de ces deux sens possibles

La deuxième étape a donc pour but de générer tous les sous-réseaux qui satisfont ces cinq conditions.

5.2.3.5 Utilisation de la résolution de systèmes linéaires

La troisième étape repose sur la résolution de systèmes linéaires. A cette étape la donnée est un réseau complètement spécifié par les réactions incluses et le sens dans lequel elles fonctionnent. La seule information manquante est le nombre de fois où chaque réaction doit être utilisée pour parvenir au bilan final. Comme le bilan final est connu et que chaque réaction incluse dans le réseau est connue, il est facile de résoudre ce problème en faisant appel à un solveur de systèmes linéaires. Comme le bilan final est bien connu et que le nombre de réactions est le plus petit possible, le système d'équations linéaires

associé n'a qu'une seule solution ou aucune (cela est garanti par construction, voir [Fan *et al.*, 2002] pour plus de détails). Le sous-réseau est donc soit validé, soit rejeté. Dans le cas où il est accepté, les coefficients d'utilisation de chaque réaction sont connus car ce sont les coefficients solutions du système linéaire posé à partir de l'ensemble des réactions du sous-réseau candidat.

5.3 Utilisation de réseaux de flux d'atomes de carbone comme abstraction pour la reconstruction métabolique

Cette approche se propose de reconstruire des voies métaboliques étant donnés les composés chimiques initiaux et finaux du réseau. On peut abstraire une voie métabolique à un réseau de réactions effectuant des opérations uniquement sur les atomes de carbones. La seule contrainte sur les réactions est alors que les nombres d'atomes de carbone en entrée et en sortie soient identiques. L'abstraction aux seuls atomes de carbone se justifie par le fait que les atomes de carbone forment le squelette des composés chimiques. Cette approche utilise cette abstraction et procède en deux étapes successives pour construire des réseaux métaboliques dont les composés en entrée et en sortie sont définis. La première étape consiste à définir des réseaux décrivant uniquement le flux des atomes de carbone dans le réseau. Le résultat de cette première étape consiste en des réseaux où seule la taille des composés, en nombre d'atome de carbone, est connue et où les réactions qui les relient sont spécifiées uniquement par le nombre d'atomes de carbone des composés qu'elles échangent. Pour chaque réseau généré à la première étape, la seconde étape consiste à ajouter des informations sur le réseau afin d'obtenir une "véritable" voie métabolique. Cela consiste à spécifier les composés chimiques impliqués, puis à déterminer les enzymes associées aux réactions du réseau construit.

Cette approche n'a été utilisée que sur deux cas ne faisant intervenir que des sucres avec un maximum de 7 atomes de carbone (voie de conversion des hexoses en pentoses (PPP) [Mittenthal *et al.*, 1998] et le cycle de Krebs [Mittenthal *et al.*, 2001]). En effet la dernière étape de la construction des réseaux est prise en charge manuellement et nécessite un grand investissement. Cela limite évidemment les applications.

L'étape automatique (la première) a pour but la construction de réseaux appelés C-nets. Les C-nets sont des réseaux décrivant le flux des atomes de carbone entre les composés. Deux types de réactions interviennent dans un C-net et à un de ces types est associé une transformation du squelette des composés. Dans un C-net, la structure chimique des composés n'est pas considérée, seul le nombre d'atomes de carbone de chaque composé

est pris en compte (voir figure 5.12).

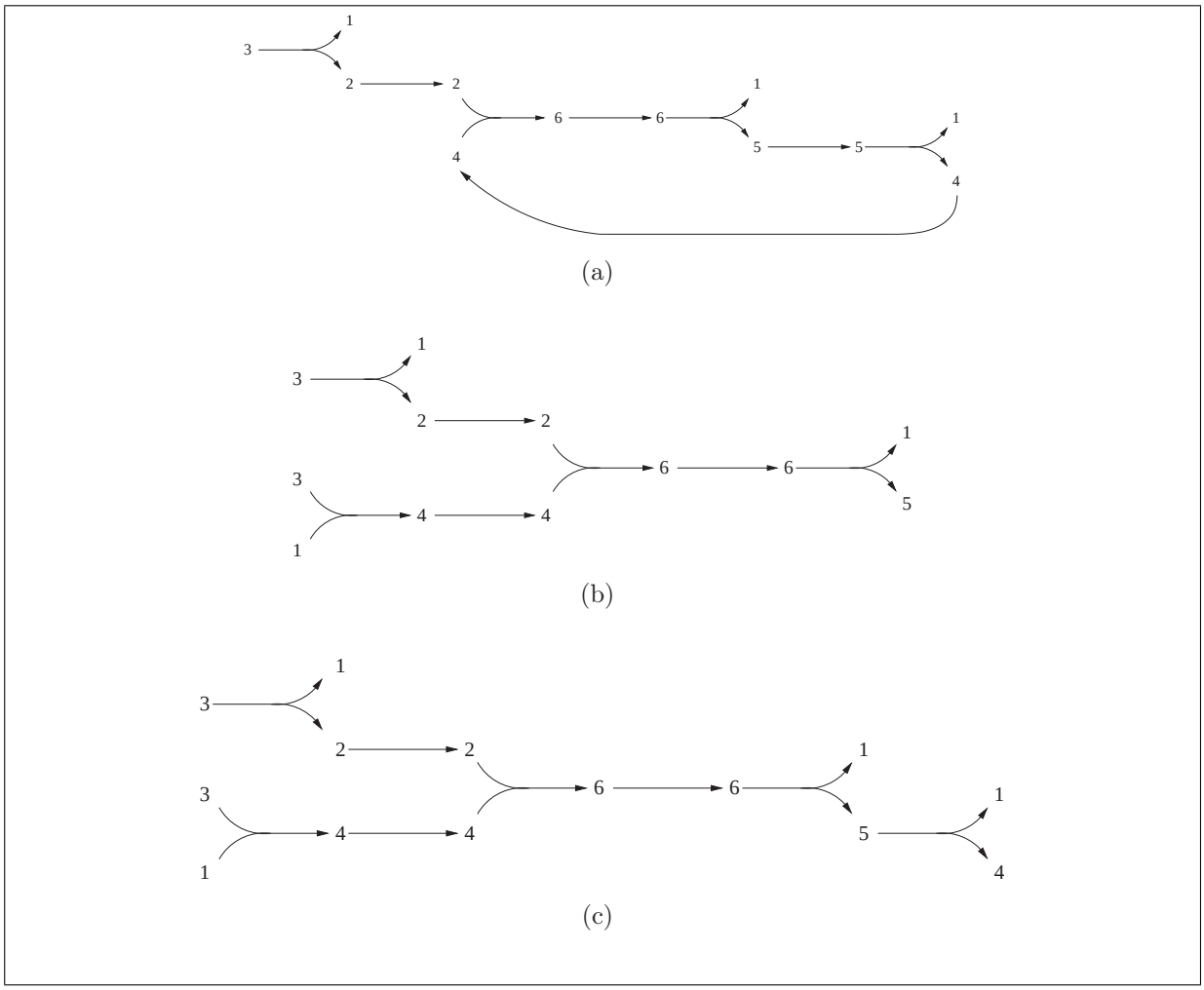


Fig. 5.12: Différents C-nets associés au cycle de Krebs - (a) convertit un pyruvate (3 carbones) en trois CO_2 (1 carbone) : $3 \rightarrow 1+1+1$, (b) convertit deux pyruvates en un 2-ketoglutarate (5 carbones) et un CO_2 : $3+3 \rightarrow 5+1$, (c) convertit deux pyruvates en un oxalacetate (4 carbones) et deux CO_2 : $3+3 \rightarrow 4+1+1$ (adapté de [Mittenthal *et al.*, 2001])

Dans la réaction $(\{r,s\},\{t,u\})$ d'un C-net, r et s représentent le nombre d'atomes de carbone des substrats et t et u ceux des produits.

Définition **11** *Réactions d'un C-net*
Il y a deux types de réactions dans un C-net :

1. *les réactions de type* $(\{x,0\},\{x,0\})$ *ont pour fonction de mimer l'action d'une enzyme ajoutant ou supprimant un groupe fonctionnel à la molécule (par exemple une phosphorylation), sans modifier le nombre de carbones*

2. *l'autre type de réaction d'un C-net est un couple de paires d'entiers positifs* $(\{r, s\}, \{t, u\})$ *qui remplit les trois conditions suivantes :*

   (a) $r + s = t + u$

   (b) *la paire* $\{r, s\}$ *est différente de la paire* $\{t, u\}$

   (c) *le nombre d'éléments égal à zéro n'excède pas 1*

   Ces trois conditions signifient respectivement :

   (a) *que le nombre total d'atomes de carbone est conservé*

   (b) *que la réaction induit effectivement une redistribution des atomes de carbone*

   (c) *que la réaction implique au moins 3 composés (1 substrat et 2 produits ou 2 substrats et 1 produit)*

   Note : la méthode ne considère que les réactions ayant au maximum 2 composés comme produits et substrats mais il aurait été tout à fait possible de considérer des n-uplets à la place des paires. Dans ce cas, seule la $3^{ème}$ *règle devrait être changée.*

Les modifications du squelette des composés se font par l'intermédiaire des réactions abstraites appelées g-réactions. Une g-réaction est une vue simplifiée ou abstraite d'une réaction chimique qui modifie le squelette des composés. Chacune des réactions du second type du C-net est associée à une réaction abstraite.

Exemple : dans la table 5.2 sont présentées toutes les classes de g-réactions associées aux réactions des C-nets dans [Mittenthal *et al.*, 1998], la règle associée et le type d'enzyme qui leur correspond.

g-réaction	Opération	règle	Type d'enzyme
1	Enlever 1 carbone	$\{x, 1\} \leftrightarrow \{x + 1, 0\}$	oxydoréductases, lyase
2	Ajouter 1 carbone	$\{x + 1, 0\} \leftrightarrow \{x, 1\}$	ligase, lyase
3	Enlever 2 carbones ou plus	$\{x, a\} \leftrightarrow \{x + a, 0\}, a \geq 2$	lyase, hydrolase, transférase
4	Ajouter 2 carbones ou plus	$\{x + a, 0\} \leftrightarrow \{x, a\}, a \geq 2$	lyase, transférase
5	Transférer 1 carbone	$\{x, y\} \leftrightarrow \{x - 1, y + 1\}$	transférase
6	Transférer 1 carbone ou plus	$\{x, y\} \leftrightarrow \{x - a, y + a\}, a \geq 1$	transférase
7	Transférer 3 carbones ou plus	$\{x, y\} \leftrightarrow \{x - a, y + a\}, a \geq 3$	transférase

Tab. 5.2: Les différentes classes de g-réactions définies dans [Mittenthal *et al.*, 1998] (adapté de [Mittenthal *et al.*, 1998])

Les deux paragraphes suivants décrivent chacun une des deux étapes de construction des réseaux suivant cette approche.

5.3.1 Construction des C-nets

Pour construire les C-nets, il faut connaître le bilan final du réseau que l'on veut construire. Par exemple, pour la reconstruction de la voie métabolique responsable de la conversion des hexoses en pentoses (PPP) (l'application visée dans [Mittenthal *et al.*, 1998]), le bilan que le réseau doit respecter est $6+6+6+6+6 \rightarrow 5+5+5+5+5+5$, c'est-à-dire la conversion de 5 hexoses en 6 pentoses.

Atomes	Paires possibles	Liste des réactions
2	$\{2,0\}$, $\{1,1\}$	$(\{2,0\},\{1,1\})$ $(\{1,1\},\{2,0\})$ $(\{2,0\},\{2,0\})$
3	$\{3,0\}$, $\{2,1\}$	$(\{3,0\},\{2,1\})$ $(\{2,1\},\{3,0\})$ $(\{3,0\},\{3,0\})$
4	$\{4,0\}$, $\{3,1\}$, $\{2,2\}$	$(\{4,0\},\{3,1\})$ $(\{4,0\},\{2,2\})$ $(\{3,1\},\{2,2\})$ $(\{3,1\},\{4,0\})$ $(\{2,2\},\{4,0\})$ $(\{2,2\},\{3,1\})$ $(\{4,0\},\{4,0\})$
5	$\{5,0\}$, $\{4,1\}$, $\{3,2\}$	$(\{5,0\},\{4,1\})$ $(\{5,0\},\{3,2\})$ $(\{4,1\},\{3,2\})$ $(\{4,1\},\{5,0\})$ $(\{3,2\},\{5,0\})$ $(\{3,2\},\{4,1\})$ $(\{5,0\},\{5,0\})$
6	$\{6,0\}$, $\{5,1\}$, $\{4,2\}$, $\{3,3\}$	$(\{6,0\},\{5,1\})$ $(\{6,0\},\{4,2\})$ $(\{6,0\},\{3,3\})$ $(\{5,1\},\{4,2\})$ $(\{5,1\},\{3,3\})$ $(\{4,2\},\{3,3\})$ $(\{5,1\},\{6,0\})$ $(\{4,2\},\{6,0\})$ $(\{3,3\},\{6,0\})$ $(\{4,2\},\{5,1\})$ $(\{3,3\},\{5,1\})$ $(\{3,3\},\{4,2\})$ $(\{6,0\},\{6,0\})$
7	$\{7,0\}$, $\{6,1\}$, $\{5,2\}$, $\{4,3\}$	$(\{7,0\},\{6,1\})$ $(\{7,0\},\{5,2\})$ $(\{7,0\},\{4,3\})$ $(\{6,1\},\{5,2\})$ $(\{6,1\},\{4,3\})$ $(\{5,2\},\{4,3\})$ $(\{6,1\},\{7,0\})$ $(\{5,2\},\{7,0\})$ $(\{4,3\},\{7,0\})$ $(\{5,2\},\{6,1\})$ $(\{4,3\},\{6,1\})$ $(\{4,3\},\{5,2\})$ $(\{7,0\},\{7,0\})$

Tab. 5.3: les différentes réactions possibles dans les C-nets faisant intervenir des composés à 7 atomes de carbone maximum

Les C-nets sont construits de manière combinatoire en générant tous les C-nets de différentes tailles respectant le bilan fixé. Les réactions disponibles pour la construction du C-net dépendent du nombre maximal d'atomes de carbone des composés. Pour un nombre maximal de 7 atomes de carbone, le nombre de réactions est de 47 (voir table 5.3).

La génération des C-nets se fait bien entendu sous la contrainte d'un nombre maximum de réactions pouvant intervenir. Dans la construction, on peut également rejeter certains réseaux pour des raisons topologiques (cycles futiles), ou toute autre condition dépendante de l'application visée. Le problème résolu dans cette première étape est donc le suivant :

PROBLÈME **5** CONSTRUCTION DE C-NETS
DONNÉES : *un ensemble R de réactions simplifiées, une contrainte stœchiométrique exprimée sous forme d'un bilan simplifié et une borne sur le nombre maximum de réactions autorisées (et éventuellement d'autres contraintes d'ordre topologique)*
RÉPONSE : *l'ensemble de tous les C-nets composés de réactions de R et vérifiant les contraintes sur le bilan et la taille finale du réseau*

L'utilisation des réactions ne décrivant que les échanges d'atomes de carbone permet de réduire le nombre de réactions à utiliser (seulement 47 pour les réactions faisant intervenir des sucres ayant jusqu'à sept atomes de carbone). Cette réduction permet de générer des réseaux de grande taille (impliquant de nombreuses réactions) car comme le nombre de g-réactions n'est pas très grand, la combinatoire sous-jacente à la construction des réseaux est moins explosive.

5.3.2 Des C-nets aux réseaux métaboliques

Une fois que tous les C-nets sont générés, il faut les contraindre pour obtenir des réseaux métaboliques. La première étape consiste à assigner à chaque réaction du C-net une g-réaction (sauf pour les réactions de type $\{x, 0\} \leftrightarrow \{x, 0\}$). Cela contraint le type d'enzyme associé à chaque réaction. Comme il peut y avoir plusieurs choix, plusieurs réseaux différents peuvent être générés à partir du même C-net.

DÉFINITION **12** *Association d'une g-réaction à une réaction d'un C-net*
A chacune des réactions du C-net, on associe une g-réaction. L'association est possible si la règle de la g-réaction est compatible avec la réaction du C-net (il peut donc y avoir plusieurs g-réactions possibles dans le cas général).

Exemple : la réaction $(\{6, 2\}, \{4, 4\})$ n'est compatible qu'avec la g-réaction 6 du tableau 5.2 ($a = 2$).

L'étape suivante consiste à assigner une molécule précise à chacun des emplacements correspondant à un composé. Le nombre de groupements fonctionnels associés aux molécules et leur type doivent être pris en compte dans cette phase pour éviter les réseaux incohérents.

Ensuite, l'intervention devient quasiment uniquement manuelle dans la mesure où il s'agit de sélectionner ces réseaux et d'en faire de véritables réseaux métaboliques putatifs. Il faut assigner effectivement à chaque réaction une enzyme si elle existe ou considérer son existence (c'est cette étape, cruciale, qui semble difficilement automatisable). Vient enfin le problème de l'évaluation des différents réseaux construits qui peut être guidée par l'utilisation de fonctions objectives (utilisation de composés énergétiques notamment).

Ce type d'approche très exploratoire ne paraît pas complètement automatisable, sauf en considérant un ensemble de réactions prédéfinies, mais dans ce cas, on perd l'avantage de la méthode qui est de pouvoir générer des réseaux complètement inattendus.

5.4 Recherche des chemins suivis par les atomes

Une réaction biochimique peut être considérée comme un échange d'atomes entre les composés impliqués dans cette réaction. Cette définition des réactions peut être utilisée dans le cadre de la reconstruction de voies métaboliques [Arita, 2000b] : en utilisant ces transferts, on peut construire un graphe où à chaque nœud est associé un atome d'un composé et où une arête est associée à un transfert d'atome dans une réaction. Un chemin dans ce graphe représente un chemin que peut suivre un atome particulier.

La connaissance de ces échanges atomiques entre composés impliqués par les réactions n'est pas une information stockée dans les différentes ressources disponibles concernant le métabolisme. Ces transferts d'atomes doivent donc être soit donnés par un expert, soit calculés. Etant donné le nombre de réactions actuellement disponibles (plusieurs milliers), il est difficilement concevable de se passer d'une méthode automatisée pour calculer ces transferts d'atomes. Dans ce but, différents algorithmes (et les programmes de résolution basés sur ces algorithmes) ont été conçus [Akutsu, 2003; Arita, 2000a]. Pour calculer ces transferts, il est nécessaire de décrire finement les composés. On les manipule donc par l'intermédiaire de graphes décrivant leur structure bidimensionnelle appelés graphes moléculaires.

Dans le paragraphe suivant la notion de graphe moléculaire est introduite. Le deuxième paragraphe se focalise sur le problème de la détermination automatique des correspondances atomiques entre composés d'une même réaction. Le dernier paragraphe donne un exemple de reconstruction de voie métabolique par cette approche.

5.4.1 Graphe moléculaire et structure bidimensionnelle des composés chimiques

Les composés chimiques sont manipulés suivant trois modes de représentation différents (voir figure 5.13) :

- la structure tridimensionnelle
- la forme plane
- le graphe moléculaire

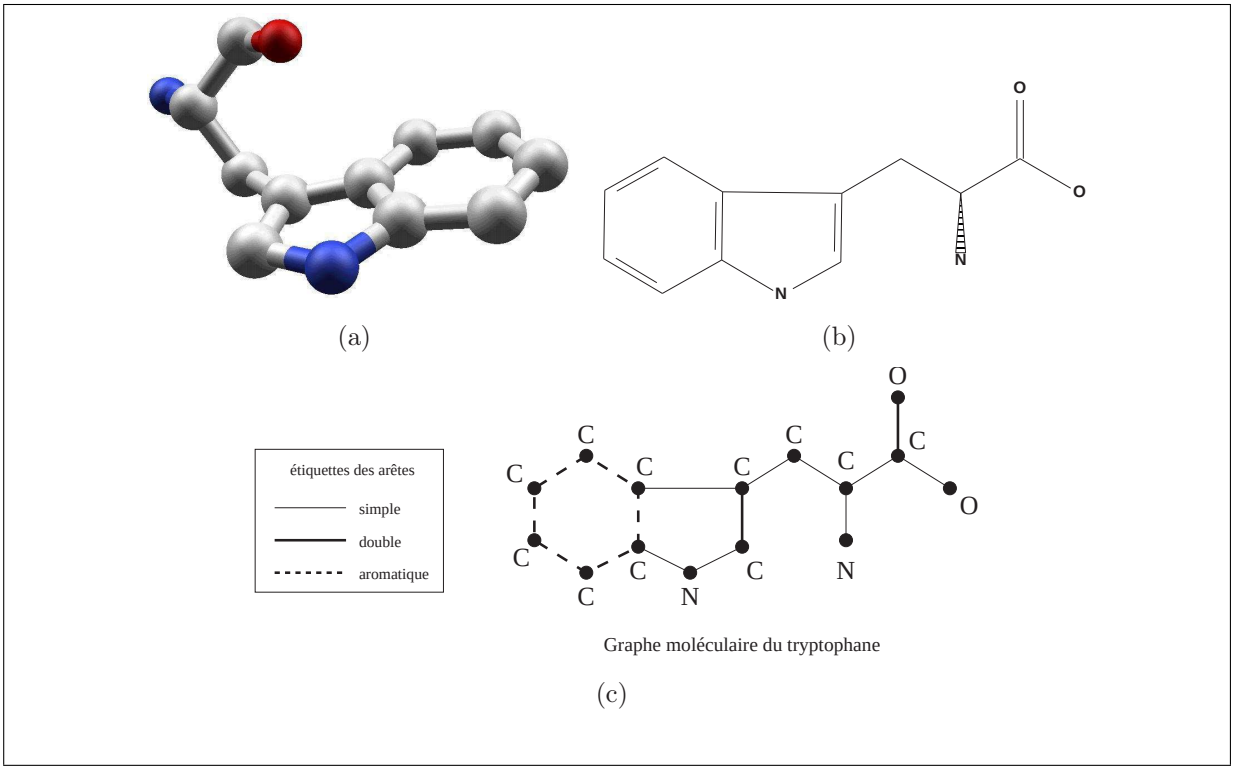


FIG. 5.13: La structure tridimensionnelle du tryptophane (a), sa représentation plane (b) et son graphe moléculaire (c)

Dans un graphe moléculaire, l'ensemble des nœuds est défini par les atomes du composé. Chaque arête correspond à une liaison chimique entre deux atomes. Il est nécessaire d'associer une étiquette aux nœuds et aux liaisons pour permettre la distinction entre les différents types d'atomes et les différents types de liaisons.

DÉFINITION **13** *Graphe moléculaire*
Le graphe moléculaire $\mathcal{GM}_C$ associé au composé C est défini par :

- *V son ensemble de nœuds, $|V|$ est égal au nombre d'atomes du composés C*
- *E son ensemble d'arêtes, $|E|$ est égal au nombre de liaisons du composé C*
- *une fonction $f_V : V \rightarrow \{C, O, H, N, P, \ldots\}$, qui permet d'étiqueter chaque nœud avec le type d'atome qui lui correspond*
- *une fonction $f_E : E \rightarrow \{simple, double, triple, aromatique\}$, qui permet d'étiqueter chaque arête avec le type de liaison qui lui correspond.*

Dans le cas du tryptophane illustré sur la figure 5.13, on remarque que les conventions graphiques de la représentation plane concernant la stéréochimie des carbones ne sont pas conservées dans le graphe moléculaire. D'une manière générale, les informations stéréochimiques (isomère optique par exemple) sont perdues dans cette représentation.

5.4.2 Calcul des transferts atomiques entre composés

En utilisant les structures des composés telles que décrites au paragraphe précédent, il est possible de poser le problème de la recherche des transferts d'atomes entre composés pour une réaction comme un problème de graphes. En observant les réactions, comme celle de la figure 5.14, on remarque que de grandes sous-structures entre les substrats et les produits sont conservées et que peu de liaisons sont altérées par la réaction (3 liaisons sont modifiées dans l'exemple).

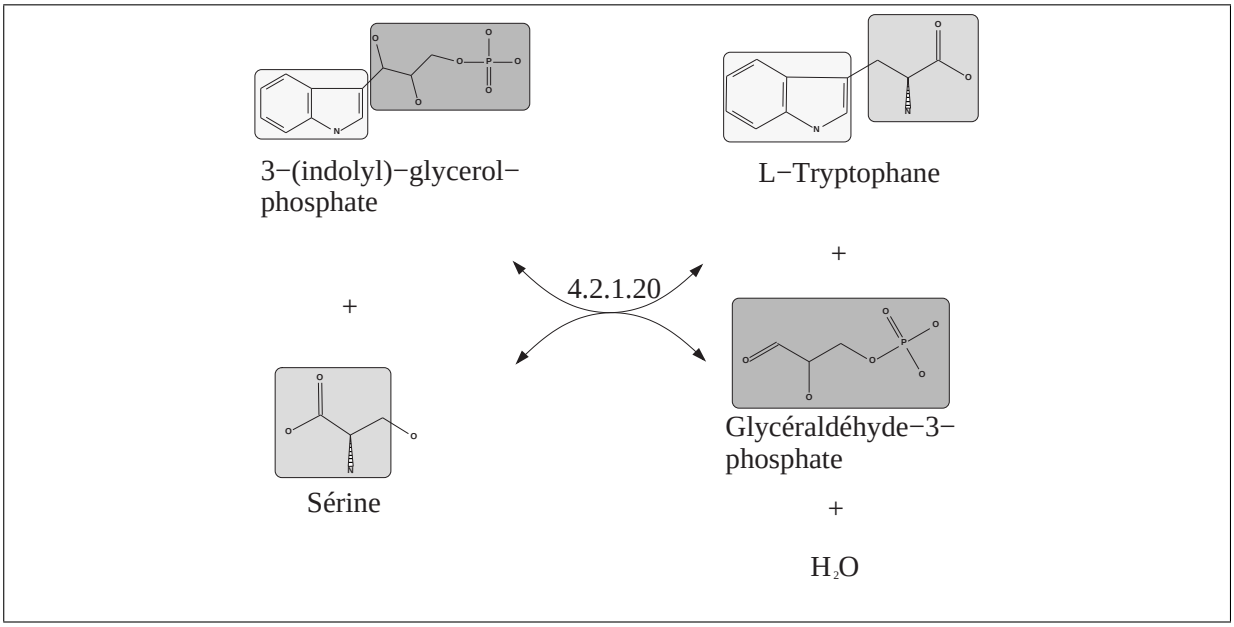


Fig. 5.14: Sous-structures inchangées par une réaction - Les sous-structures entourées de chaque côté de la réaction ne sont pas modifiées par la réaction

Ces indications nous invitent à définir le problème de la recherche des transferts d'atomes entre composés d'une réaction comme un problème d'optimisation. Le problème est de trouver une correspondance atomique qui conserve au maximum la topologie des graphes moléculaires. Le critère à optimiser dans ce problème est celui de la minimisation du nombre d'arêtes à détruire. Le problème à résoudre est un problème d'optimisation bien identifié, appelé Sous-graphe Commun Maximal, connu pour être $\mathcal{NP}$-difficile [Crescenzi and Kann, 1998, GT46].

Définition **14** *Graphes isomorphes*
Deux graphes $\mathcal{G}_1 = (V_1, E_1)$ *et* $\mathcal{G}_2 = (V_2, E_2)$ *sont isomorphes si et seulement si il existe une bijection* $\phi : V_1 \rightarrow V_2$ *telle que :*

$$\forall (u, v) \in E_1 \quad (\phi(u), \phi(v)) \in E_2$$
et
$$\forall (w, x) \in E_2 \quad (\phi^{-1}(w), \phi^{-1}(x)) \in E_1$$

PROBLÈME **6** SOUS-GRAPHE COMMUN MAXIMAL
DONNÉES : *deux graphes* $\mathcal{G}_1 = (V_1, E_1)$ *et* $\mathcal{G}_2 = (V_2, E_2)$
RÉPONSE : $E_1^{'} \subseteq E_1$ *et* $E_2^{'} \subseteq E_2$ *tels que* $\mathcal{G}_1^{'} = (V_1, E_1^{'})$ *et* $\mathcal{G}_2^{'} = (V_2, E_2^{'})$ *sont isomorphes*
MESURE : $\left|E_1^{'}\right|$
OPTIMISATION : max

Dans le cas des graphes moléculaires, comme les nœuds des graphes sont étiquetés cela modifie légèrement la définition de deux graphes isomorphes, il faut ajouter la condition que les étiquettes des nœuds mis en correspondance doivent être égales (on pourrait également imposer que les étiquettes des arêtes soient identiques, mais comme les deux méthodes présentées après n'en tiennent pas compte, la définition donnée les ignore également) :

DÉFINITION **15** *Graphes étiquetés isomorphes*
Deux graphes $\mathcal{G}_1 = (V_1, E_1)$ *et* $\mathcal{G}_2 = (V_2, E_2)$ *sont isomorphes si et seulement si il existe une bijection* $\phi : V_1 \rightarrow V_2$ *telle que :*

$$\begin{aligned}
&\forall (u,v) \in E_1 \quad (\phi(u), \phi(v)) \in E_2 \\
&\qquad\qquad\quad et\ f_V(u) = f_V(\phi(u)) \\
&\qquad\qquad\quad et\ f_V(v) = f_V(\phi(v)) \\
&et \\
&\forall (w,x) \in E_2 \quad (\phi^{-1}(w), \phi^{-1}(x)) \in E_1 \\
&\qquad\qquad\quad et\ f_V(w) = f_V(\phi^{-1}(w)) \\
&\qquad\qquad\quad et f_V(x) = f_V(\phi^{-1}(x))
\end{aligned}$$

Deux algorithmes sont présentés qui résolvent spécifiquement le problème de l'assignation des correspondances atomiques induites par une réaction. Le premier algorithme présenté résout le problème de manière exacte. Sa complexité est exponentielle (problème $\mathcal{NP}$-difficile) mais dans certains cas particuliers elle devient polynômiale. Le second algorithme est une résolution heuristique basée sur la recherche des plus grandes sous-structures communes entre deux composés.

5.4.2.1 Résolution exacte pour des cas particuliers de réactions en temps polynômial

Une formalisation originale mais équivalente du problème du SOUS-GRAPHE COMMUN MAXIMAL est donnée dans [Akutsu, 2003]. Bien que restant $\mathcal{NP}$-difficile dans le cas général, ce problème peut être résolu en temps polynômial, lorsqu'il porte sur certains types de réactions.

Formulation du problème Dans [Akutsu, 2003], une opération particulière sur les graphes représentant les molécules est introduite et appelée *coupe chimique*.

Définition **16** *Coupe chimique (adapté de [Akutsu, 2003])*
Soit un graphe $\mathcal{G}(V, E)$ et un entier positif non nul c. Une coupe chimique est une partition du graphe $\mathcal{G}$ en composantes connexes obtenues en retirant au plus c arêtes qui satisfait la condition suivante :

soit le graphe $\widetilde{\mathcal{G}}\left(\widetilde{V}, \widetilde{E}\right)$ où :

- *$\widetilde{V}$ est l'ensemble des composantes connexes obtenues*
- *$(c_i, c_j) \in \widetilde{E}$ si et seulement si il y a eu suppression d'une arête entre les composantes c_i et c_j de $\widetilde{V}$*

le graphe $\widetilde{\mathcal{G}}$ est une étoile

Exemple : la figure 5.16 illustre cette définition

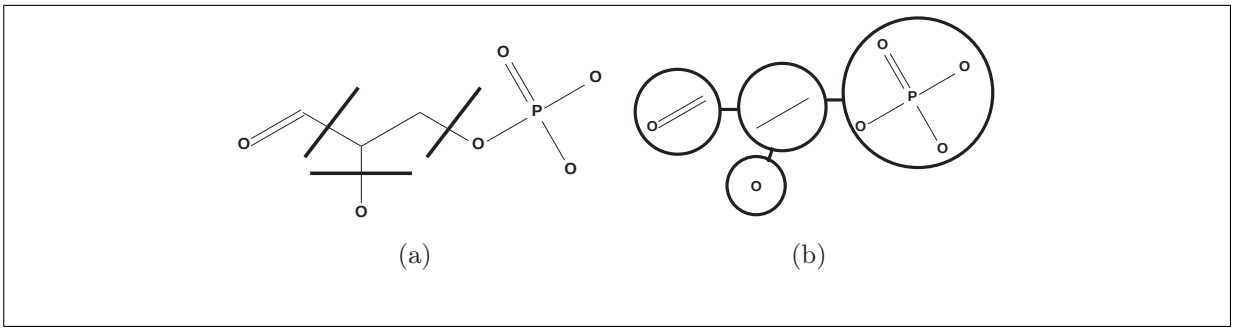


Fig. 5.15: Exemple d'une coupe chimique de taille 3 (a) et du graphe étoile $\widetilde{\mathcal{G}}$ associé (b) pour le glycéraldéhyde-3-phosphate

Cette définition est utilisée dans la définition du problème de base.

Problème **7** Recherche des correspondances atomiques par coupes chimiques et isomorphismes de graphes *(adapté de [Akutsu, 2003])*
Données : *une réaction chimique $S_1 + \ldots + S_p \leftrightarrow P_1 + \ldots + P_q$, où $S_1, \ldots, S_p$ et $P_1, \ldots, P_q$ sont des graphes moléculaires et où le multi-ensemble des atomes de $S_1, \ldots, S_p$ est égal au multi-ensemble des atomes de $P_1, \ldots, P_q$ (i.e. la loi de conservation est satisfaite), et un entier c.*

Réponse : *un ensemble de coupes chimiques de taille c pour chacun des $S_1, \ldots, S_p$ et $P_1, \ldots, P_q$ tel que le multi-ensemble des composantes connexes des S_i obtenu par les coupes chimiques est égal au multi-ensemble des composantes connexes des P_i obtenu (le test d'égalité est basé sur l'isomorphisme des composantes connexes (définition 15))*

Exemple : la figure suivante donne un exemple de solution de ce problème pour une réaction simple.

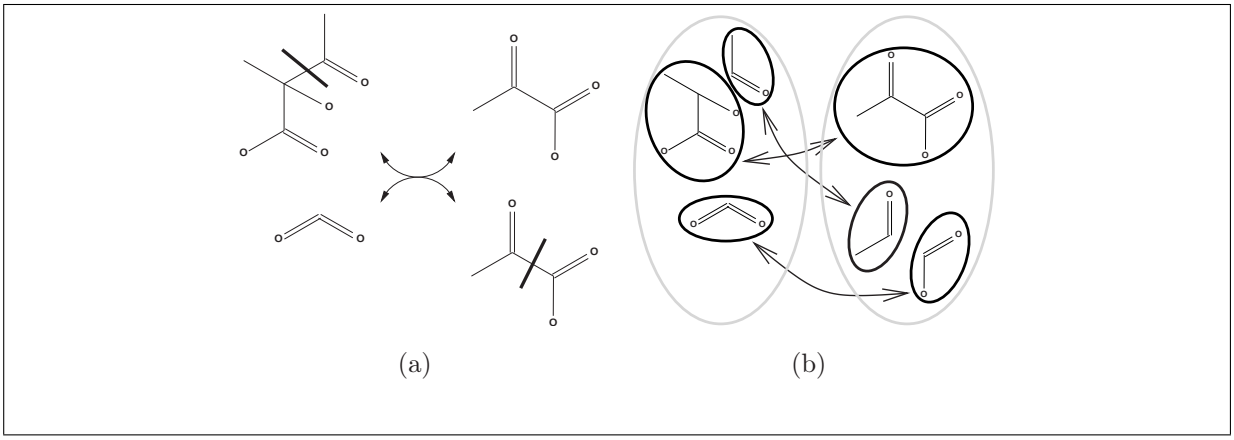


Fig. 5.16: Illustration du problème Recherche des correspondances atomiques par coupes chimiques et isomorphismes de graphes - Les coupes chimiques sont présentées en (a) et l'égalité entre les deux multi-ensembles (qui utilise l'isomorphisme de graphes) en (b)

Pour chacune des paires de composantes connexes isomorphes, on dispose d'une fonction bijective entre les nœuds de ces composantes. A la fin de l'exécution de l'algorithme, on dispose donc d'une fonction bijective entre l'ensemble des nœuds du graphe des substrats et l'ensemble des nœuds du graphe des produits. Si on supprime les arêtes correspondantes à celles supprimées lors des coupes chimiques, on a bien deux graphes, un pour chaque côté de la réaction, qui sont isomorphes.

Résultats de complexité Le problème précédent est $\mathcal{NP}$-complet même pour $p = q = 2$. La $\mathcal{NP}$-complétude est montrée en utilisant une réduction à partir du Problème Des Mariages (3-Dimensional Matching) [Garey and Johnson, 1979].

Un algorithme ayant une complexité en $\mathcal{O}\left(n^{1,5}\right)$ existe pour le cas particulier où $p = q = 2$, $c = 1$ et où les graphes moléculaires des composés sont des arbres. La restriction des composés à des arbres permet d'utiliser un algorithme linéaire pour le test d'isomorphisme. Cet algorithme utilise un algorithme de recherche de cycle de longueur 4 dans un graphe dont les nœuds sont les composantes connexes obtenues par les coupes.

L'algorithme esquissé ci-dessus n'est efficace que dans certain cas particuliers, dans le cas général, sa complexité le rend, comme toute autre résolution exacte du problème Sous-graphe Commun Maximal, inutilisable. Aussi, est-il nécessaire d'envisager l'utilisation d'heuristiques pour le cas général, afin de bénéficier de temps d'exécution raisonnables. Le paragraphe suivant présente un tel algorithme.

5.4.2.2 Algorithme glouton basé sur la recherche du Sous-graphe Induit Commun Connexe Maximal

L'algorithme se base sur la recherche du plus grand sous-graphe connexe induit entre deux graphes (Sous-graphe Induit Commun Connexe Maximal). Ce problème est décrit en détail à l'annexe B.

Les données de l'algorithme consiste en une réaction chimique $S_1 + \ldots + S_p \leftrightarrow P_1 + \ldots + P_q$, où $S_1, \ldots, S_p$ et $P_1, \ldots, P_q$ sont décrits par leurs graphes moléculaires et où le multi-ensemble des atomes de $S_1, \ldots, S_p$ est égal au multi-ensemble des atomes de $P_1, \ldots, P_q$ (*i.e.* la loi de conservation doit être satisfaite).

Comme le principe de l'algorithme est simple, nous nous contentons de l'illustrer sur un exemple. Il s'agit à chaque pas de l'algorithme de :

1. choisir la plus grande sous-structure commune entre les substrats et les produits
2. de considérer que la correspondance atomique partielle induite par cette sous-structure fait partie de la solution finale
3. de supprimer les deux sous-graphes induits par cette sous-structure des graphes des substrats et des produits
4. de revenir à l'étape 1 tant qu'il subsiste des graphes substrats et produits

La figure 5.17 montre un exemple d'exécution de l'algorithme.

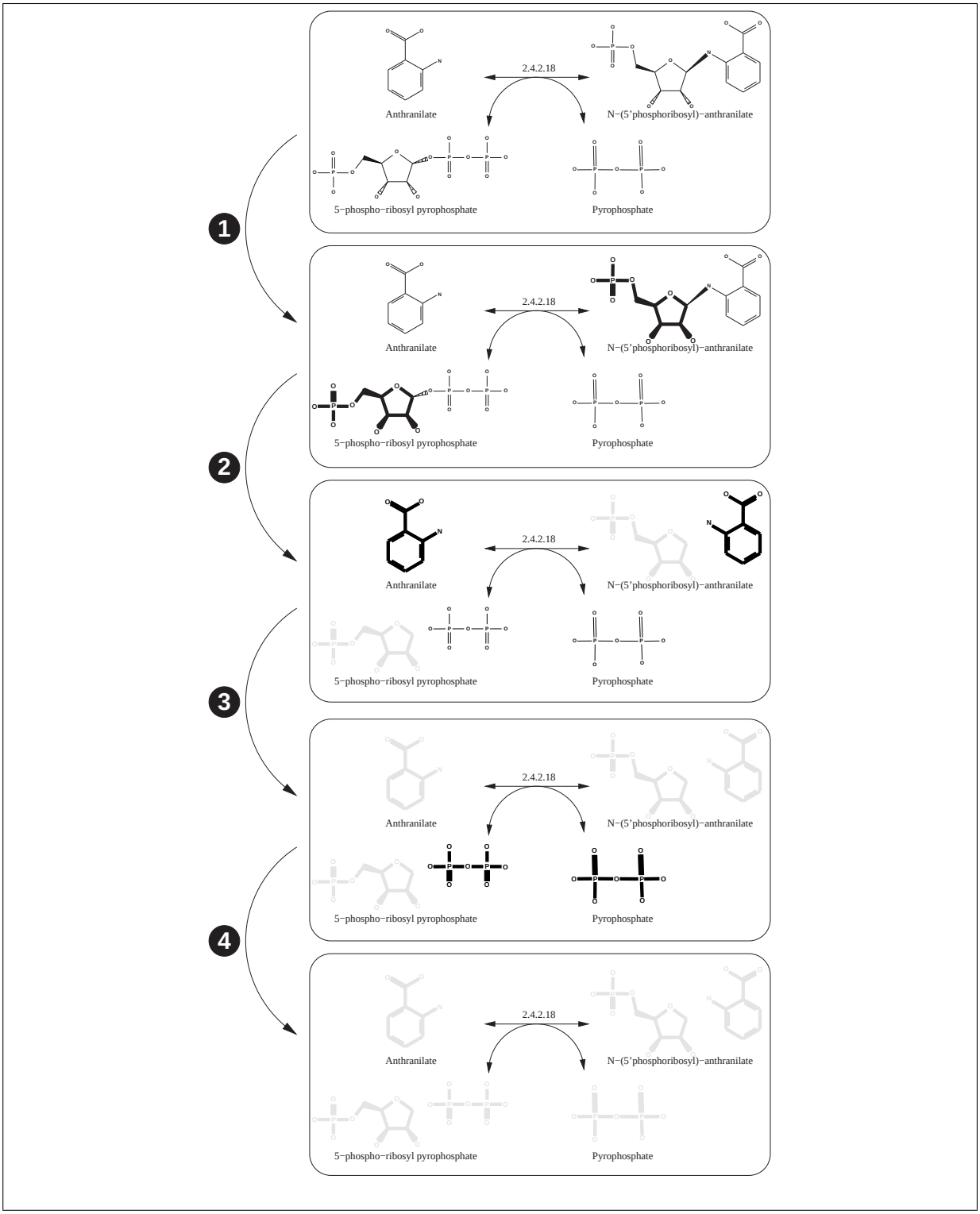


Fig. 5.17: Un exemple d'exécution de l'algorithme glouton pour le calcul des correspondances atomiques dans une réaction - Une arête est perdue de chaque côté de la réaction ce qui correspond à la réalité biologique et à la solution optimale du problème Sous-graphe Commun Maximal

La résolution du problème SOUS-GRAPHE COMMUN MAXIMAL fournit un isomorphisme (ou plusieurs car il n'y a pas forcément une seule solution) entre les atomes des composés de chaque côté de la réaction ce qui permet de construire le graphe des transferts d'atomes.

5.4.3 Recherche de chemins métaboliques

A partir de tous les isomorphismes calculés pour chaque réaction, il est maintenant possible de construire le graphe représentant l'intégralité des transferts d'atomes entre les composés impliqués dans le métabolisme. Dans ce graphe, chaque nœud est associé à un atome d'un composé. Chaque arête correspond à un transfert d'un atome entre deux composés. Chaque arête est étiquetée par la réaction qui induit le transfert. La figure 5.18 donne un exemple d'un tel graphe pour une réaction.

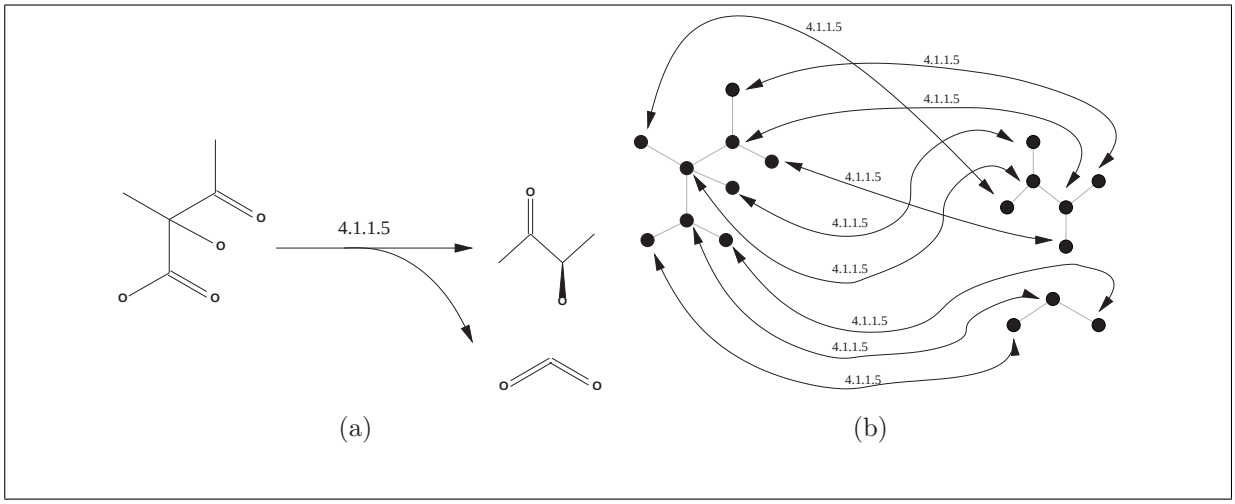


FIG. 5.18: Un exemple de réaction (a) et le graphe des transferts d'atomes induit par cette réaction (b)

N'importe quel chemin reliant deux atomes dans le graphe des transferts d'atomes correspond à un enchaînement de réactions qui conduit à un échange d'au moins un atome entre le composé de départ et le composé d'arrivée. Ce type de chemins avec échange d'au moins un atome, comme le montre la figure 5.19, peut correspondre à des chemins métaboliques identifiés. L'idée est donc, étant donnés deux composés x et y (la requête) de rechercher tous les chemins de x à y dans le graphe des transferts. Comme il est impossible d'énumérer tous les chemins possibles, la solution adoptée dans [Arita, 2000b] est de rechercher les k plus courts chemins en terme de nombre de réactions impliquées (avec l'algorithme décrit dans [Eppstein, 1998]).

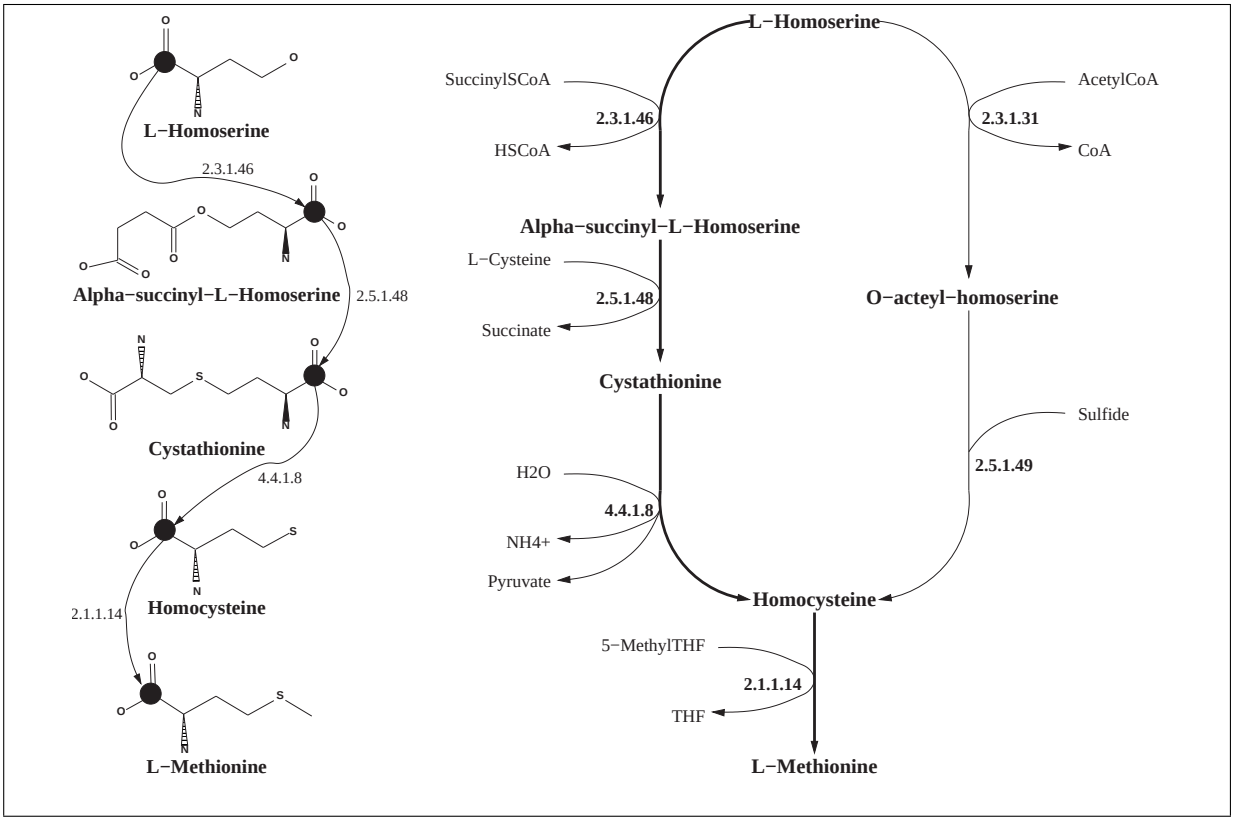


Fig. 5.19: Le chemin suivi par un atome peut correspondre à une voie métabolique

Les deux précédents chapitres ont permis d'introduire les différentes méthodes de reconstruction de voies métaboliques qui se basent respectivement sur :

- la connaissance de voies métaboliques pour des organismes modèles,
- la connaissance de l'ensemble des réactions potentiellement impliquées dans le réseau à reconstruire,
- la connaissance des transferts atomiques entre les composés impliqués dans l'ensemble des réactions

Chacune de ces méthodes présente des avantages et des inconvénients, les rendant complémentaires dans la tâche de reconstruction de voies métaboliques.

Chapitre 6

Propriétés topologiques et génomiques des réseaux métaboliques

La connaissance des propriétés des réseaux métaboliques, tant au niveau des réactions qu'à celui de l'organisation des gènes impliqués dans la catalyse de ces réactions, est une donnée importante dans le processus de la reconstruction. Il apparaît donc primordial de s'intéresser à ces deux aspects.

Ce chapitre se décompose en deux parties. La première est consacrée aux caractéristiques topologiques des graphes métaboliques et comprend une partie consacrée aux caractéristiques des trois modèles de graphes principaux utilisés comme références pour analyser les graphes métaboliques. La seconde partie se concentre sur les caractéristiques génomiques des voies métaboliques, c'est-à-dire sur l'organisation sur le chromosome bactérien des gènes dont les produits sont impliqués dans des voies métaboliques.

6.1 Caractéristiques topologiques des réseaux métaboliques

La caractérisation de la topologie des graphes construits à partir de réseaux métaboliques permet d'avoir une vue d'ensemble de l'architecture de ces réseaux et d'extraire des informations sur leur organisation.

Afin de les caractériser, ces graphes sont confrontés à des modèles dont les caractéristiques sont bien connues. Les trois paragraphes suivants portent sur la définition de ces caractéristiques puis sur leur comportement pour les trois modèles de graphes de référence et enfin sur les caractéristiques des graphes métaboliques.

6.1.1 Quelques mesures caractéristiques des graphes

Les caractéristiques les plus étudiées des graphes sont la distribution de l'arité des nœuds, la valeur du coefficient d'agrégation et le diamètre.

6.1.1.1 Distribution de l'arité des nœuds

Le degré d'un nœud (ou arité) est le nombre d'arêtes qui sont reliées à ce nœud.

Dans un graphe orienté, la distinction peut être faite entre le degré inférieur (nombre d'arcs arrivant au nœud) et le degré supérieur (nombre d'arcs partant du nœud).

6.1.1.2 Coefficient d'agrégation

Le coefficient d'agrégation (*clustering coefficient*) a été introduit dans [Watts and Strogatz, 1998]. Il mesure la probabilité qu'un nœud quelconque d'un graphe forme une clique avec ses voisins immédiats.

Soit $v(\mu)$ les voisins immédiats du nœud μ dans le graphe $\mathcal{G} = (V, E)$. Le nombre maximum de connexions possibles entre ces nœuds est égal à $|v(\mu)|\,(|v(\mu)| - 1)/2$. Si seulement y_μ arêtes existent effectivement entre les $|v(\mu)|$ nœuds voisins de μ, alors C_μ, le coefficient d'agrégation du nœud μ est :

$$C_\mu = \frac{2y_\mu}{|v(\mu)|\,(|v(\mu)| - 1)} \tag{6.1}$$

La moyenne des C_μ sur tous les nœuds d'un graphe est le coefficient d'agrégation du graphe. Ce coefficient est donc égal à :

$$C_{\mathcal{G}=(V,E)} = \frac{1}{|V|}\sum_{\mu\in V} C_\mu = \frac{1}{|V|}\sum_{\mu\in V} \frac{2y_\mu}{|v(\mu)|\,(|v(\mu)| - 1)} \tag{6.2}$$

Pour un arbre, le coefficient d'agrégation est $C_{arbre} = 0$, pour un graphe complet, il est de valeur maximale $C_{complet} = 1$.

6.1.1.3 Diamètre d'un graphe

Dans un graphe $\mathcal{G} = (V, E)$, la distance ℓ_{uv} entre deux nœuds u et v de V peut être définie comme le nombre d'arêtes qui forment le chemin de longueur minimum entre ces deux nœuds.

Suivant les auteurs, le diamètre peut être :

– la moyenne des distances pour chaque paire de nœuds du graphe notée $\bar{\ell}_{\mathcal{G}=(V,E)}$

– la distance maximale entre toutes paires de nœuds du graphe notée $\ell^{+}_{\mathcal{G}=(V,E)}$

De manière générale, pour un graphe $\mathcal{G} = (V, E)$, on a :

$$\bar{\ell}_{\mathcal{G}=(V,E)} = \frac{1}{|V|(|V|-1)} \sum_{(i,j)\in V^2 \wedge i \neq j} \ell_{ij} \tag{6.3}$$

$$\ell^{+}_{\mathcal{G}=(V,E)} = \max\left(\ell_{ij}\right), \forall (i,j) \in V^2 \wedge i \neq j \tag{6.4}$$

6.1.2 Modèles de graphes

Ces caractéristiques ont été très largement étudiées pour quelques modèles de graphes. Les modèles aléatoires, *small-world* et *scale-free* sont présentés ici. Les paragraphes suivants sont adaptés de [Albert, 2001; Albert and Barabási, 2002; Dorogovtsev and Mendes, 2001].

6.1.2.1 Modèle aléatoire

L'ouvrage de référence concernant les graphes aléatoires et leurs caractéristiques est [Bollobás, 1985].

Description et construction Pour définir un graphe aléatoire, il suffit de préciser :

- le nombre m de nœuds
- le nombre n d'arêtes

Pour générer un graphe correspondant à ce modèle, il suffit de choisir aléatoirement pour chaque arête quelle paire de nœuds, parmi les $\frac{m(m-1)}{2}$ possibles, elle relie.

Une façon proche de définir ce modèle est de donner :

- le nombre m de nœuds
- la probabilité p qu'une arête relie une paire de nœuds (p est alors égal à $\frac{2n}{m(m-1)}$ avec les paramètres de la définition précédente)

Une instance de ce modèle est obtenue en prenant successivement toutes les paires de nœuds possibles du graphe et en les reliant par une arête avec une probabilité p. Le nombre attendu $E(n)$ d'arêtes est fonction de m et p : $E(n) = p \times \frac{m(m-1)}{2}$.

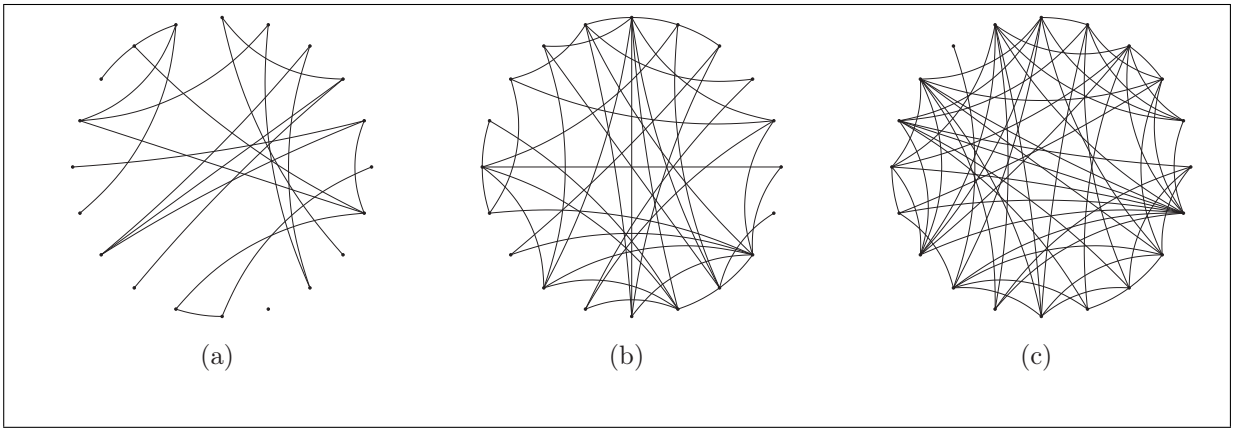


FIG. 6.1: Exemples de graphes aléatoires avec $m = 20$ nœuds et (a) $n = 20$, (b) $n = 40$, (c) $n = 60$ arêtes

Caractéristiques

Distribution de l'arité des nœuds Dans le cas d'un graphe aléatoire $\mathcal{G}(m, p)$, où m est le nombre de nœuds et p la probabilité qu'une paire de nœuds soit reliée par une arête, le degré d_i du nœud i suit une distribution binomiale de paramètres $m - 1$ et p :

$$P(d_i = d) = C^d_{(m-1)} p^d (1 - p)^{m-1-d} \tag{6.5}$$

où p^d est la probabilité qu'un nœud ait un degré égal à d. La probabilité de l'absence d'arête supplémentaire est $(1 - p)^{m-1-d}$. De plus, il y a $C^d_{(m-1)}$ façons de sélectionner les nœuds associés par ces arêtes.

Si X_d est le nombre de nœuds de degré d, et $P(X_d = r)$ la probabilité que le nombre de nœuds du graphe ayant un degré de d soit égal à r, on a

$$E(X_d) = mP(d_i = d) = \lambda_d \tag{6.6}$$

avec

$$\lambda_d = mC^d_{(m-1)} p^d (1 - p)^{m-1-d} \tag{6.7}$$

La distribution des degrés suit approximativement une distribution binomiale :

$$P(d) = C^d_{m-1} p^d (1 - p)^{m-1-d} \tag{6.8}$$

Lorsque la probabilité p que deux nœuds soient reliés par une arête est très petite $p \ll 1$ (c'est-à-dire $n \ll m$) alors $P(X_d = r)$ tend vers une distribution poissonnienne

$$P(X_d = r) \simeq \exp -\lambda_d \frac{\lambda^r_d}{r!} \tag{6.9}$$

De plus dans le cas où $p \ll 1$ (c'est-à-dire $n \ll m$) la distribution du degré des nœuds peut être remplacée par une distribution de Poisson

$$P(d) \simeq \exp^{-pm\frac{(pm)^d}{d!}} = \exp^{-\bar{d}\frac{(\bar{d})^d}{d!}} \tag{6.10}$$

où $\bar{d}$ est le degré moyen du graphe aléatoire.

Coefficient d'agrégation Soit un nœud quelconque dans un graphe aléatoire et ses voisins immédiats, la probabilité que deux de ses voisins soient reliés par une arête est égale à la probabilité que deux nœuds sélectionnés aléatoirement soient reliés. En conséquence, le coefficient d'agrégation d'un graphe aléatoire est :

$$C_{aléatoire} = \frac{\bar{d}}{m} \tag{6.11}$$

Diamètre Il y a approximativement $\bar{d}^{\ell}$ nœuds à distance ℓ ou inférieure de n'importe quel nœud. Grâce à des simulations numériques, lorsque le nombre de nœud m est égal à $\bar{d}^{\bar{\ell}}$, on observe que le diamètre du graphe aléatoire approche l'expression suivante :

$$\bar{\ell} \simeq \frac{\ln m}{\ln \bar{d}} \tag{6.12}$$

6.1.2.2 Modèle *small-world*

Ce modèle a pour but de reproduire dans les graphes construits les propriétés des réseaux décrivant les relations sociales entre individus (*Small-World problem*) [Watts and Strogatz, 1998].

Description et construction Ces graphes sont caractérisés par :

- un coefficient d'agrégation élevé. C'est-à-dire que si un nœud A est relié à un nœud B et le même nœud B est relié à un nœud C, alors il y a de fortes chances pour que le nœud A soit relié au nœud C
- la longueur du chemin le plus court entre deux nœuds est toujours faible

La construction d'un graphe suivant ce modèle est la suivante :

1. On construit un treillis en anneau contenant m nœuds et où chaque nœud est relié à ses $2k$ plus proches voisins
2. Chaque arête allant d'un nœud à ses k plus proches voisins dans le sens horaire est réaffectée avec une probabilité p en choisissant aléatoirement (distribution uniforme) un nouveau nœud destination

Les paramètres nécessaires sont donc, outre le nombre de nœuds m :

– la distance k des plus proches nœuds voisins qui sont reliés par une arête à un nœud du graphe
– la probabilité p qu'une arête soit réaffectée

Ce modèle permet donc de quantifier grâce à p le degré de désordre introduit. Si p est nul le graphe à une structure régulière. Au contraire, lorsque p est égal à 1, le graphe est totalement désordonné (mais pas complètement équivalent à un graphe aléatoire $\mathcal{G}(m, \frac{km}{2})$ [Barrat and Weigt, 2000]).

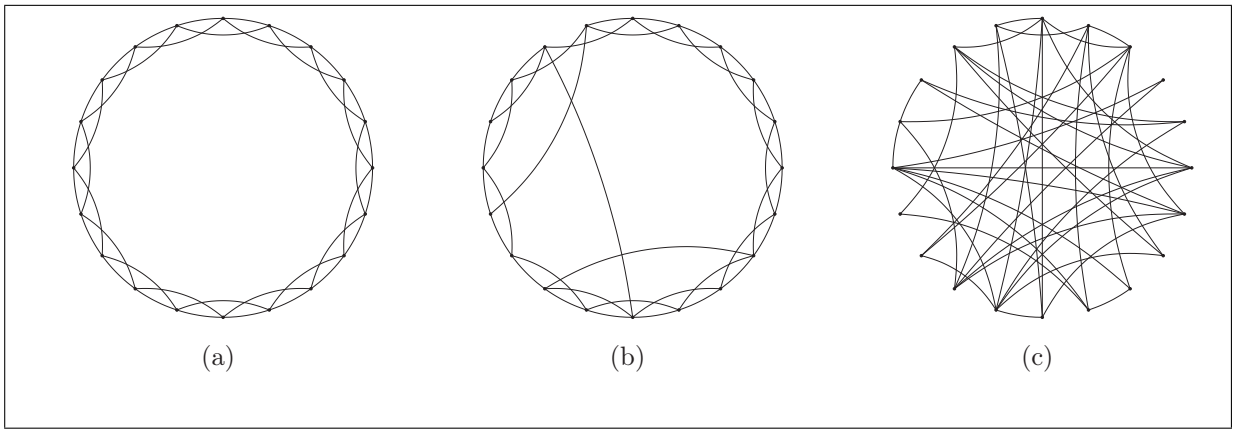


FIG. 6.2: Exemples de graphes *small world* obtenus par la méthode de construction énoncée précédemment pour $k = 2$, $m = 20$ et (a) $p = 0$, (b) $p = 0.1$, (c) $p = 1$ (adapté de [Barrat and Weigt, 2000])

Caractéristiques

Distribution de l'arité des nœuds Ce paragraphe est basé sur [Barrat and Weigt, 2000].

Si $p \neq 0$, pmk arêtes sont réaffectées (sur mk arêtes totales), mais le nombre d'arêtes et donc la moyenne de l'arité des nœuds restent inchangée $\bar{d} = 2k$.

D'après la méthode de construction, sur les $2k$ arêtes initiales dont le nœud i est une extrémité, après la réaffectation des arêtes, un minimum de k arêtes sont toujours reliées au nœud i. Le degré du nœud i peut dont être noté $d_i = k + n_i$, où $n_i \geq 0$. n_i peut être décomposé en deux parties : les $n_i^1 \leq k$ arêtes qui ont été laissées en place (à chaque fois avec une probabilité $(1 - p)$), et les $n_i^2 = n_i - n_i^1$ liens reconnectés au nœud i (avec une probabilité pour chacun de $\frac{p}{m}$).

On a donc

$$P(n_i^1) = C_{n_i^1}^k (1-p)^{n_i^1} p^{k-n_i^1} \tag{6.13}$$

et

$$P(n_i^2) = \frac{kp^{n_i^2}}{n_i^2!} \exp(-pk), \quad \text{pour } m \gg 0 \tag{6.14}$$

finalement on obtient,

$$P(d) = \sum_{n=0}^{\min(d-k,k)} C_n^k (1-p)^n p^{k-n} \frac{(kp)^{d-k-n}}{(d-k-n)!} \exp(-pk), \quad d \geq k. \tag{6.15}$$

Coefficient d'agrégation Le point de départ du modèle est un treillis circulaire à une dimension. Chacun des m nœuds est connecté à ses $2k$ plus proches voisins (où $k \geq 2$). Dans un tel réseau le coefficient d'agrégation ne dépend pas du nombre de nœuds mais de la topologie du réseau (donc de k). Sur les $(2k)((2k)-1)/2 =$ nœuds voisins d'un nœuds, il y en a $3k(k-1)/2$ qui sont reliés entre eux ce qui conduit à

$$C(k,0) = \frac{3(k-1)}{2(2k-1)} \tag{6.16}$$

Pour $p > 0$ la probabilité que deux voisins du nœud i qui étaient connectés pour $p = 0$ (c'est-à-dire dans l'anneau) soient toujours voisins de i et soient toujours connectés est de $(1-p)^3$ car il faut conserver 3 arêtes intactes.

Définissons $\tilde{C}$ comme le rapport entre le nombre moyen d'arêtes entre les voisins d'un nœud et le nombre moyen d'aêtes possibles entre ses voisins. Cette définition est équivalente à

$$\tilde{C}(k,p) = \frac{3 \times \text{nombre de triangles}}{\text{nombre de triplets connectés}} \tag{6.17}$$

Alors pour un modèle *small-world* $\tilde{C}(p)$ vaut

$$\tilde{C}(k,p) = \frac{3(k-1)}{2(2k-1)}(1-p)^3 \tag{6.18}$$

La déviation entre $\tilde{C}(k,p)$ et $C(k,p)$ est faible et est de l'ordre de $1/m$ [Barrat and Weigt, 2000].

On pose donc

$$C(k,p) \sim \tilde{C}(k,p) = C(k,0)(1-p)^3 \tag{6.19}$$

Diamètre La distance moyenne entre nœuds a tendance à décroître dramatiquement même pour des valeurs de p relativement faible, ceci étant dû aux *courts circuits* introduits dans la phase de réaffectation des arêtes [Watts and Strogatz, 1998].

Pour $p = 0$, la distance moyenne correspond à la distance moyenne d'un treillis en anneau. Contrairement au cas du coefficient d'agrégation le nombre de nœuds m intervient dans la longueur moyenne entre les nœuds $\bar{\ell}(m, p)$, qui dans le cas du treillis en anneau est

$$\bar{\ell}(m, 0) = \frac{m(m + k - 1)}{4k(m - 1)} \sim \frac{m}{4k} \tag{6.20}$$

Dans le cas général, il apparaît que $\bar{\ell}(m, p)$ est de la forme suivante

$$\bar{\ell}(m, p) \sim m^* F_k(\frac{m}{m^*}) \tag{6.21}$$

avec $F_k(u \ll 1) \sim u$, $F_k(u \gg 1) \sim \ln u$ et $m^* \sim p^{-1}$,

Pour plus de détails voir [Barrat and Weigt, 2000]. D'autres travaux [Walsh, 1999; Kleinberg, 2000; Puniyani *et al.*, 2001] s'intéressent à l'impact de la topologie des graphes de type *small-world* sur la recherche de chemins dans ces graphes.

6.1.2.3 Modèle *scale-free*

D'après des études empiriques (résumées dans [Albert and Barabási, 2002]), il apparaît que bon nombre de réseaux de grandes tailles (comme Internet, les réseaux de transports aériens, les réseaux de collaborations scientifiques...) ont tendance à suivre les caractéristiques du modèle dit *scale free* dont la principale caractéristique est que la distribution des degrés des nœuds suit une loi exponentielle, c'est-à-dire que la probabilité $P(d)$ qu'un nœud du graphe ait un degré égal à d suit $P(d) \sim d^{-\gamma}$, où γ est le paramètre de cette loi de distribution.

Description et construction Le modèle suit deux principes fondamentaux :

- Le graphe grandit par l'addition à chaque étape de nouveaux nœuds
- L'adjonction des nouvelles arêtes favorise l'attachement à des nœuds dont le degré est élevé

Le modèle est ainsi défini par les paramètres suivants :

- m_0 est le nombre initial de nœuds
- m ($m \leq m_0$) le degré de chaque nouveau nœud qui est ajouté lors de l'étape de croissance du graphe
- m_{max} ($m_{max} \geq m_0$) le nombre final de nœuds

Un graphe suivant ce modèle est construit avec l'algorithme suivant :

1. *initialisation* : $t = 0$;
   Un graphe $\mathcal{G}_0$ avec m_0 nœuds et aucune arête est construit

2. *croissance* : $t = t + 1$;
   un nouveau nœud est ajouté avec $m(\leq m_0)$ arêtes le reliant à m nœuds déjà existants suivant la règle énoncée ci-dessous

3. *attachement préférentiel* :
   Les m nœuds auxquels le nouveau nœud est relié sont choisis en fonction de leur degré d_i. La probabilité que le nœud i de degré d_i soit relié au nouveau nœud vaut :

$$P(d_i) = \frac{d_i}{\sum_j d_j} \tag{6.22}$$

4. tant que $t \leq (m_{max} - m_0)$ retour à l'étape 2

Après t itérations, le nombre de nœuds est égal à $m_{max} = m_0 + t$ et le nombre d'arêtes à tm.

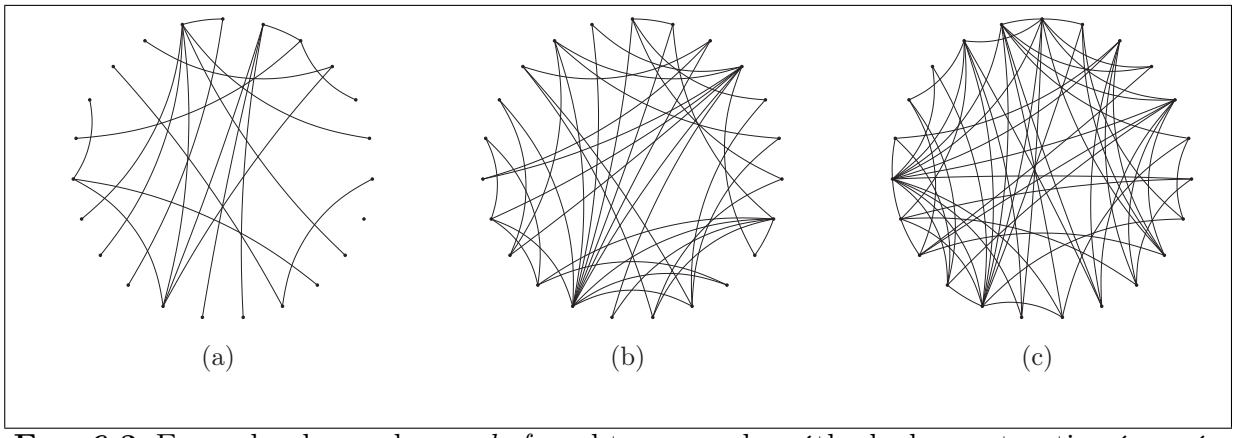


Fig. 6.3: Exemples de graphes *scale free* obtenus par la méthode de construction énoncée précédemment pour $m_0 = 3$, $m_{max} = 20$ et (a) $m = 1$, (b) $m = 2$, (c) $m = 3$

Les variantes du modèle qui ne suivent que l'un de ces deux principes (*Croissance* ou *Attachement préférentiel*) ne possèdent pas la même distribution de degré des nœuds [Barabási *et al.*, 1999]. Dans le même travail, il est montré analytiquement que quelque soit les paramètres m et m_0 du modèle, lorsque t tend vers l'infini alors $P(d) \sim d^{\gamma}$ avec $\gamma = 3$.

Caractéristiques

Distribution de l'arité des nœuds La particularité du modèle dit *scale free* se rapporte justement à la distribution des degrés des nœuds. Celle-ci suit une loi exponentielle $P(d) \sim d^{-\gamma}$ avec $\gamma = 3$.

Coefficient d'agrégation Il n'y a apparemment aucune expression analytique donnant le coefficient de *clustering* des graphes respectant le modèle *scale free*. Néanmoins, des simulations semblent montrer que le coefficient est 5 fois supérieur à celui d'un graphe aléatoire et suit la loi $C = m^{-0.75}$ [Albert, 2001].

Diamètre Il n'y a apparemment aucune expression analytique donnant la longueur moyenne des plus courts chemins entre les nœuds d'un graphe de type *scale free*.

Cependant, des simulations numériques semblent montrer que $\bar{\ell}_{scale\ free}$ est toujours inférieur à $\bar{\ell}_{aléatoire}$.

De plus, toujours grâce à des simulations numériques, il a été mis en évidence que $\bar{\ell}_{scale\ free}$ croit logarithmiquement avec le nombre de nœuds.

6.1.3 Caractéristiques des graphes métaboliques

Les paragraphes suivants présentent les travaux portant sur les caractéristiques topologiques des réseaux métaboliques.

6.1.3.1 Travaux et graphes associés

[Jeong *et al.*, 2000; Podani *et al.*, 2001; Ravasz *et al.*, 2002] se basent sur les réseaux métaboliques de 43 organismes complètement séquencés issues de [Overbeek *et al.*, 2000]. La construction du graphe est la suivante : pour chaque organisme ayant un réseau métabolique composé de r réactions catalysées par e enzymes impliquant c composés, un graphe à $r + e + c$ nœuds a été construit. Chaque nœud correspondant à une enzyme est relié aux nœuds correspondant aux réactions qu'elle catalyse. Les nœuds correspondant aux substrats et aux produits d'une réaction sont reliés au nœud correspondant à cette réaction.

Dans [Fell and Wagner, 2000; Wagner and Fell, 2001] est étudié le réseau métabolique de la bactérie *Escherichia coli*. Les réactions ont été sélectionnées manuellement sur la base de ressources textuelles. Le réseau comporte 317 réactions et 287 molécules. Deux types de graphes ont été construits à partir de ces données : un graphe des réactions $\mathcal{G}_R$ (semblable à celui de la figure 3.1(c)) et un graphe des composés où deux nœuds représentant deux composés sont reliés si les composés interviennent dans la même réaction (que le composé soit substrat ou produit de la réaction n'importe pas). Deux versions de chaque graphe ont été construites, l'une en tenant compte de tous les composés, l'autre en ignorant les composés secondaires suivants : ATP, ADP, NAD, NADP, NADH, NADPH, CO_2, NH_3, SO_4, thioredoxin, phosphate et pyrophosphate.

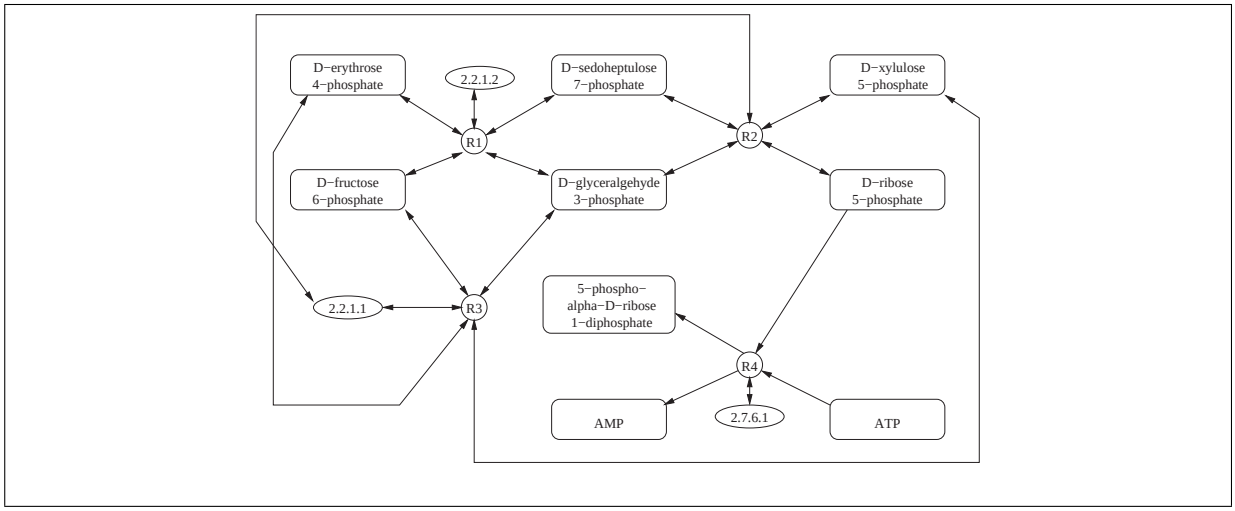


FIG. 6.4: Graphe construit à partir de 4 réactions (dont 1 irréversible : R_4) suivant la méthode de [Jeong *et al.*, 2000; Podani *et al.*, 2001; Ravasz *et al.*, 2002] (adapté de [Jeong *et al.*, 2000])

Enfin, dans [Ma and An-Ping, 2003], 80 graphes métaboliques ont été construits pour 80 organismes dont la séquence du génome est complètement disponible. La procédure de construction a pour but de construire un graphe des métabolites $\mathcal{G}_M$ (du type de celui de la figure 3.2(b)). L'intégralité des étapes de construction a été faite manuellement, sauf la sélection des réactions présentes pour chaque organisme qui est tirée de [Kanehisa and Goto, 2000]. Les réactions sont celles de la banque LIGAND [Goto *et al.*, 2002].

Bien que les différents travaux traitant de la topologie des réseaux métaboliques n'utilisent pas la même construction de graphes, des caractéristiques générales des réseaux métaboliques ont été dégagées comme la topologie générale, la longueur moyenne des plus courts chemins entre composés et l'importance de certains composés.

6.1.3.2 Topologie générale

Quelque soit la méthode de construction adoptée, les caractéristiques topologiques des graphes métaboliques (comme de nombreux autres graphes de grandes tailles tel que le web, les réseaux sociaux, les réseaux de distribution d'électricité, le système nerveux de *C.elegans*) semble indiquer une bonne adéquation avec le modèle *scale free* (et à moindre proportion avec le modèle *small world*), c'est-à-dire ayant pour principales caractéristiques une faible longueur moyenne des plus courts chemins entre deux nœuds et une distribution exponentielle de l'arité des nœuds.

6.1.3.3 Influence de la méthode de construction sur la longueur moyenne des chemins entre composés

L'inclusion, lors de la phase de construction, des composés secondaires conduit à la création de court-circuits à l'intérieur du graphe construit et fait considérablement chuter la longueur moyenne des plus courts chemins entre les nœuds d'un graphe : cette valeur oscille autour de 3,5 dans [Jeong *et al.*, 2000] contre une valeur de 8 dans [Ma and An-Ping, 2003]. Le fait de considérer la réversibilité des réactions lors de la construction du graphe a également une influence non négligeable sur la longueur des chemins entre deux nœuds.

6.1.3.4 Composés les plus impliqués dans les réseaux

Il y a plusieurs moyens d'ordonner une liste des composés impliqués dans les réseaux métaboliques en se basant sur les graphes construits : [Jeong *et al.*, 2000; Ouzounis and Karp, 2000] les trient en fonction du degré de leur nœud associé, [Wagner and Fell, 2001] en fonction du degré de leur nœud associé et de la longueur moyenne des chemins séparant leur nœud associé à tous les autres nœuds, [Ma and An-Ping, 2003] en fonction du nombre d'organismes dans lequel chaque composé possède un nœud associé.

De ces analyses, il ressort que si l'on ne tient pas compte, lors de la construction, des composés primaires et secondaires, les composés les plus souvent sélectionnés sont l'ATP, l'ADP, NAD, NADP, NADH, NADPH, CO_2, NH_3, H_2O ... Si ces derniers composés sont ignorés lors de la construction, il apparaît que les composés qui sont systématiquement en tête de liste interviennent principalement dans la glycolyse et le cycle de Krebs.

6.1.3.5 Autres caractéristiques et observations

Un autre travail [Podani *et al.*, 2001] suggère, sur la base d'une classification hiérarchique des graphes représentant les réseaux métaboliques de bactéries, d'archæbactéries et d'eucaryotes, que les réseaux métaboliques des archæbactéries sont plus proches des réseaux métaboliques d'eucaryotes que des réseaux métaboliques de bactéries.

Au moins deux travaux ([Ravasz *et al.*, 2002; Gagneur *et al.*, 2003]) se sont intéressés à l'application de méthodes de classification hiérarchique sur les nœuds des graphes métaboliques, les classes correspondant à des groupes de nœuds formant des sous-graphes à fort coefficient d'agrégation. Il en ressort que les classes qui sont constituées tendent à rassembler des composés souvent impliqués dans les mêmes voies métaboliques.

6.2 Organisation génomique des réseaux métaboliques

Chez les bactéries, l'organisation des gènes en opérons (voir § 1.2.2) joue un grand rôle dans la régulation transcriptionnelle. Il n'est donc pas étonnant d'observer qu'une bonne partie des gènes codant pour les enzymes d'une voie métabolique sont souvent présents au sein d'un même opéron. Les deux parties suivantes rapportent les résultats de travaux visant à d'une part préciser cette observation et d'autre part à en tirer partie pour la prédiction d'opérons dans les organismes procaryotes.

6.2.1 Enzymes et organisation chromosomique

Un moyen pour étudier l'impact sur l'organisation chromosomique des voies métaboliques est de mesurer et de comparer pour deux gènes codant pour des enzymes, leur "distance réactionnelle" et leur "distance chromosomique". La distance réactionnelle est déterminée à partir du réseau métabolique de l'organisme (supposé connu) et donne une idée de la distance entre les deux réactions catalysées par les deux gènes. La distance chromosomique renseigne, elle, sur la proximité chromosomique entre les deux gènes.

Définition **17** *Distance réactionnelle*
Soit un graphe métabolique $\mathcal{G}_R$ où chaque nœud est associé à une réaction et où deux réactions sont reliées si elles ont en commun un composé[3]*. Soient deux gènes g_1 et g_2 catalysant respectivement les réactions r_1 et r_2 associées aux nœuds n_1 et n_2. La distance réactionnelle $d_r(g_1, g_2)$ entre les deux gènes g_1 et g_2 est la longueur du plus court chemin entre les nœuds n_1 et n_2 dans $\mathcal{G}_R$.*

Note : si le graphe métabolique est orienté, alors cette mesure n'est plus une distance car $d_r(g_1, g_2) \neq d_r(g_2, g_1)$

Définition **18** *Distance chromosomique*
Soit un chromosome C portant deux gènes g_1 et g_2, la distance chromosomique $d_c(g_1, g_2)$ entre deux gènes g_1 et g_2 est le nombre de gènes les séparant plus un. Si le chromosome est circulaire, c'est le nombre minimum de gènes les séparant qui est pris en compte.

Sur la base de ces définitions, il est possible d'étudier la corrélation entre ces deux mesures pour toutes les paires d'enzymes d'un organisme. Une telle étude a été menée pour *Escherichia coli* dans [Rison *et al.*, 2002].

[3] un tel graphe est montré sur la figure 3.1(c), sa construction est décrite au paragraphe § 3.1. Se reporter au paragraphe § 6.1.3.3 pour apprécier l'influence de la méthode de construction sur la longueur des chemins

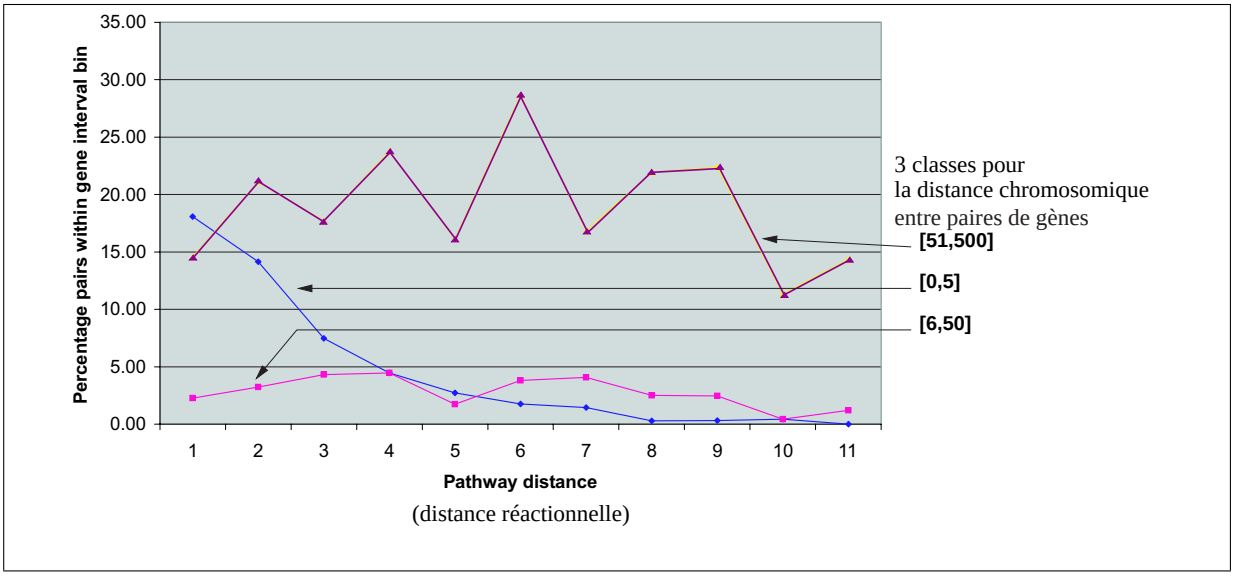


FIG. 6.5: Distance réactionnelle entre paires d'enzymes en fonction de la distance chromosomique les séparant (extrait de [Rison *et al.*, 2002])

Pour toutes les paires de gènes codant pour des enzymes, la distance chromosomique et la distance réactionnelle ont été calculées. La figure 6.5 illustre une partie de ces résultats. Pour trois intervalles différents de la distance chromosomique $[0, 5]$, $[6, 50]$ et $[51, 500]$, la distribution de la distance réactionnelle est tracée. Les paires d'enzymes pour lesquelles la distance chromosomique est faible ont une plus grande propension à être également proche en terme de distance réactionnelle. Lorsque la distance chromosomique augmente, la distribution de la distance réactionnelle tend vers une distribution en cloche.

Sur la base de ces observations, il est possible d'affirmer que chez *Escherichia coli*, il existe (et cela n'est pas une surprise) une pression pour que des gènes impliqués dans le fonctionnement d'une même voie métabolique soient co-localisés sur le chromosome. Cela pousse à rechercher, pour prédire des opérons à l'échelle d'organismes entiers, des groupes d'enzymes codées par des gènes co-localisés sur les chromosomes et catalysant des réactions voisines dans les réseaux métaboliques.

6.2.2 Recherche d'opérons à partir de voies métaboliques

A l'exception d'*Escherichia coli* [Salgado *et al.*, 2001; Karp *et al.*, 2002c] et de *Bacillus subtilis* [Itoh *et al.*, 1999], les bornes des unités de transcription (et donc des opérons) sur les génomes bactériens sont rarement connues de manière précise.

Il existe plusieurs façon de prédire les opérons (voir [Yada *et al.*, 1999; Ermolaeva *et al.*, 2000; Salgado *et al.*, 2000; Zheng *et al.*, 2002], mais une des particularités des gènes présents au sein d'un même opéron est qu'ils sont le plus souvent fonctionnellement liés comme c'est le cas pour les gènes qui codent pour des enzymes impliquées dans une même

voie métabolique. Un moyen d'apprécier la participation à la même fonction biologique pour des gènes codant pour des enzymes est de mesurer leur distance réactionnelle (décrite dans le paragraphe précédent).

Un moyen de prédire des opérons est donc de rechercher des gènes, codant pour des enzymes, co-orientés et co-localisés sur le chromosome dont les réactions associées sont proches sur le réseau métabolique de l'organisme. Un exemple d'opéron lié à une voie métabolique est montré sur la figure 6.6. Un tel groupe de gènes correspond donc au type de résultat recherché.

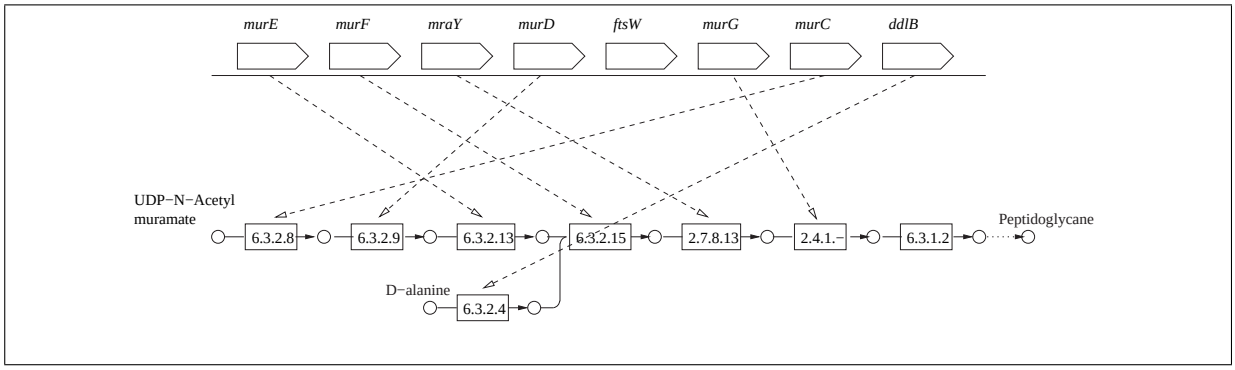


Fig. 6.6: Schéma représentant une portion du génome d'*Escherichia coli* regroupant des gènes qui codent pour des enzymes catalysant des réactions qui se succèdent - les gènes *murE*, *murF*, *mraY*, *murD*, *murG*, *murC* et *ddlB* codent pour des enzymes qui catalysent des réactions impliquées dans la biosynthèse du peptidoglycane (extrait de [Nakaya *et al.*, 2001])

Le problème de la prédiction d'opérons à l'aide d'un réseau réactionnel a été abordé dans [Ogata *et al.*, 2000] et [Zheng *et al.*, 2002] sans toutefois être formellement posé. Dans les deux cas, seul l'algorithme utilisé pour la résolution est donné. Dans les paragraphes suivants, un exemple est présenté qui permet de mieux comprendre le problème à résoudre puis les deux algorithmes sont présentés, suivi de quelques résultats de prédictions obtenus sur des génomes complets par [Zheng *et al.*, 2002].

6.2.2.1 Définition informelle du problème

Les deux illustrations de la figure 6.6, qui reprennent le cas de la figure 6.7, permettent de se faire une assez bonne idée du problème posé. Dans le cas (a), 3 groupes de gènes sont définis car la voie métabolique correspond à 3 groupes de gènes sur le chromosome. Dans le cas de (b), on autorise un trou (le gène *ftsW*) sur les groupes des gènes sur le chromosome, on obtient alors un seul groupe de gènes et de réactions. Le même type de relaxation est possible pour le graphe réactionnel.

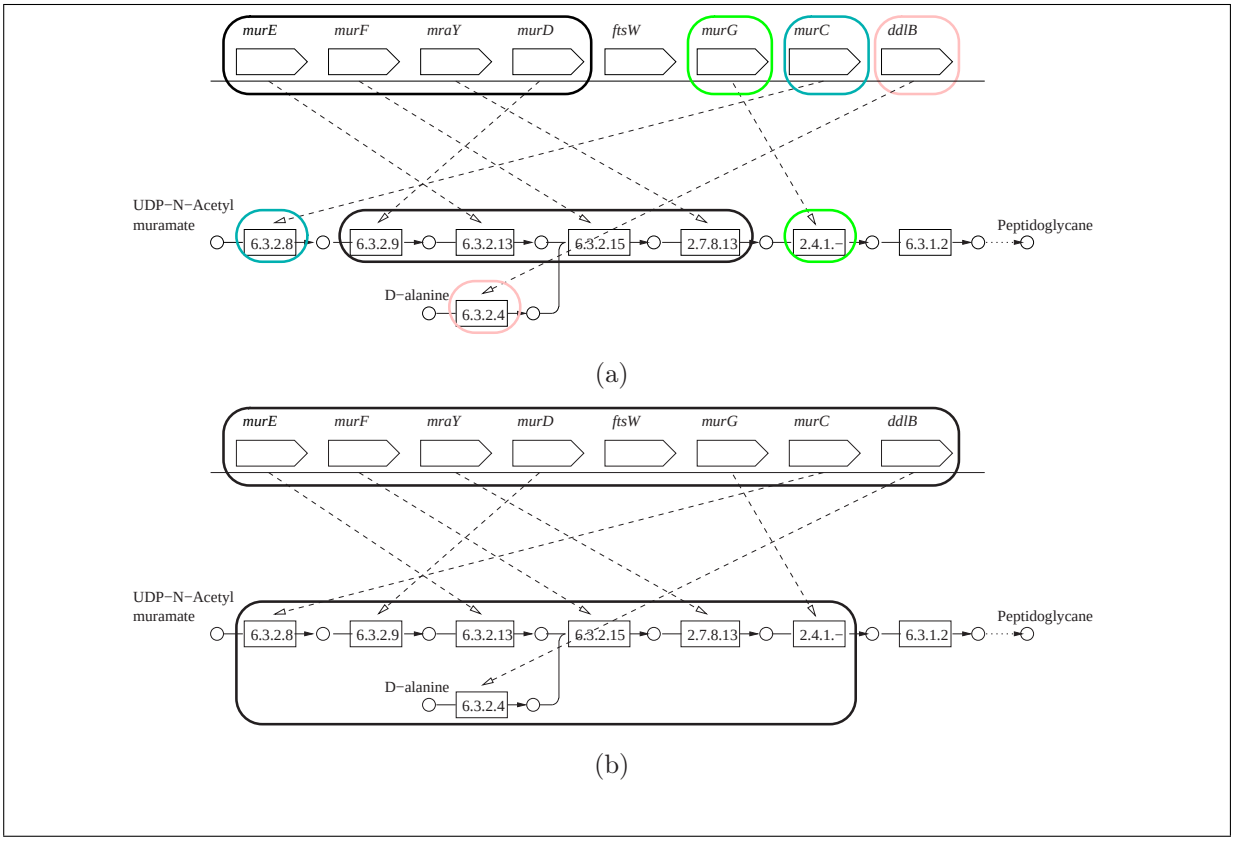


FIG. 6.7: Recherche d'opérons à partir de voies métaboliques pour les graphes de la figure 6.6 avec (a) aucune relaxation et (b) une relaxation sur la distance chromosomique entre les gènes d'un même groupe

Les groupes de gènes adjacents et co-orientés correspondent dans les deux cas à des groupes de réactions reliées entre elles, il s'agit de trouver les groupes de la plus grande taille possible.

Dans les deux parties suivantes, nous présentons les algorithmes utilisés dans [Ogata *et al.*, 2000] et [Zheng *et al.*, 2002].

6.2.2.2 Algorithme de [Ogata *et al.*, 2000]

La figure 6.8 montre le principe de l'algorithme proposé dans [Ogata *et al.*, 2000] et qui est le suivant :

initialisation : chaque couple de nœuds en correspondance est un groupe

1. on recherche deux groupes de nœuds qui sont co-localisés sur les deux graphes (on considère que deux nœuds sont "co-localisés" sur le graphe g s'ils sont à une distance maximale de $1+\delta_g$, δ_g étant un paramètre de l'algorithme)
2. ces groupes de nœuds sont marqués comme faisant partie du même groupe
3. recommencer l'étape 1 tant que la partition n'est pas stable

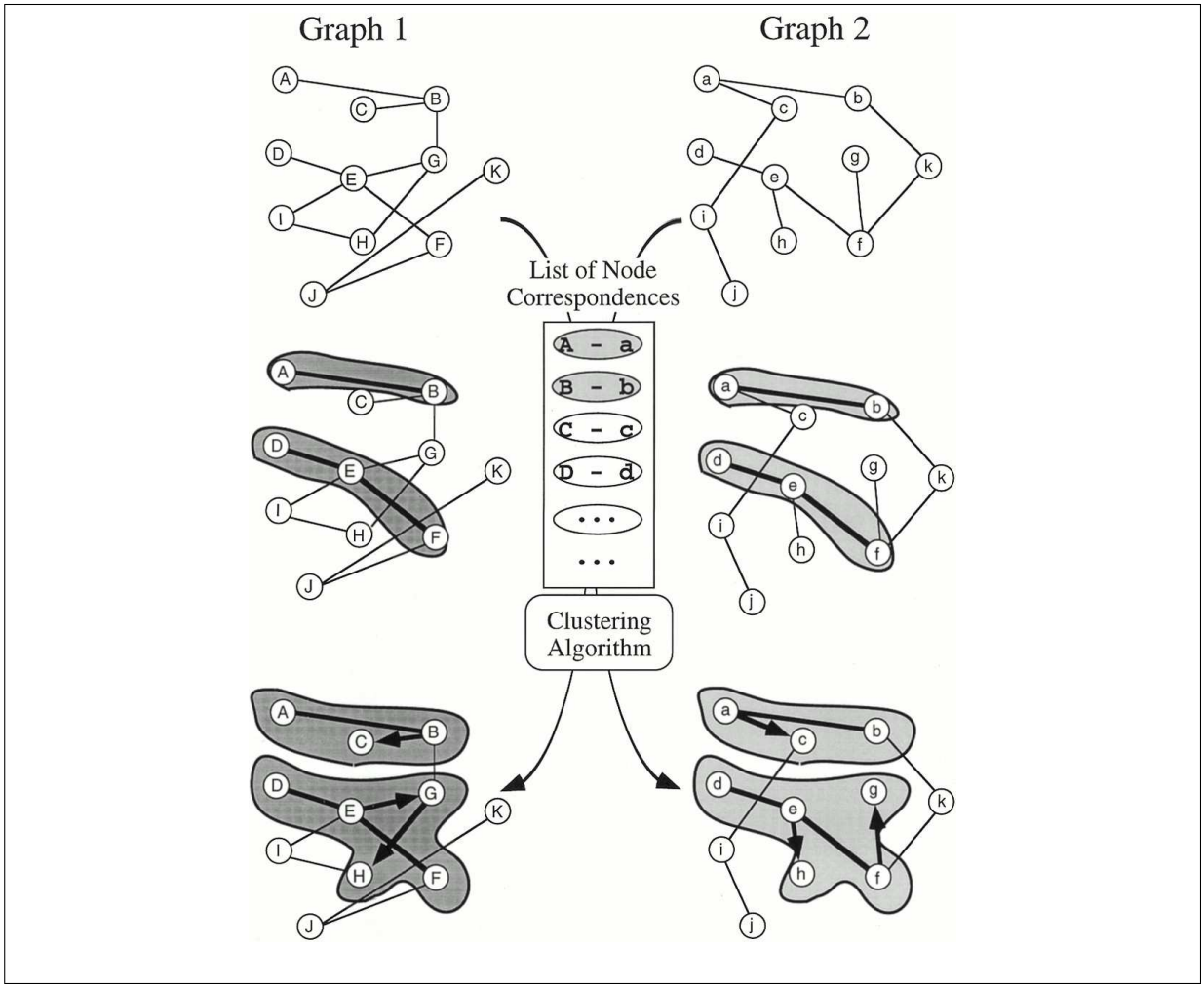


Fig. 6.8: Principe de l'algorithme - Etant donnés deux graphes et une liste de correspondance entre les nœuds des graphes, on forme des groupes de nœuds de plus en plus gros en y ajoutant successivement des nouveaux nœuds (extrait de [Ogata *et al.*, 2000])

6.2.2.3 Algorithme de [Zheng *et al.*, 2002]

L'algorithme décrit dans [Zheng *et al.*, 2002], et illustré sur la figure 6.9 consiste en :
– pour tous les nœuds d'un des deux graphes faire

1. pour le nœud choisi, trouver la liste de ses voisins à distance maximale de n, n étant un paramètre de l'algorithme
2. aller rechercher les nœuds correspondants dans l'autre graphe
3. si ces nœuds forment une composante connexe alors l'ensemble des nœuds est sélectionné

Une seconde étape consiste à fusionner les opérons prédits qui sont co-orientés et successifs ou qui se chevauchent, comme illustré sur la figure 6.10, afin d'obtenir des opérons de plus grandes tailles.

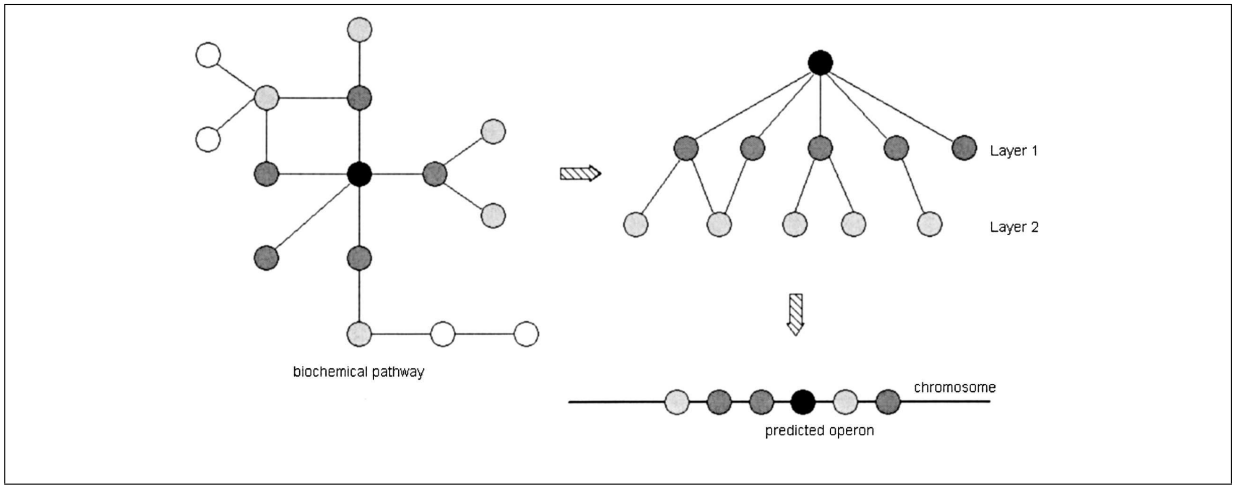


Fig. 6.9: Principe de l'algorithme de [Zheng *et al.*, 2002] (extrait de [Zheng *et al.*, 2002])

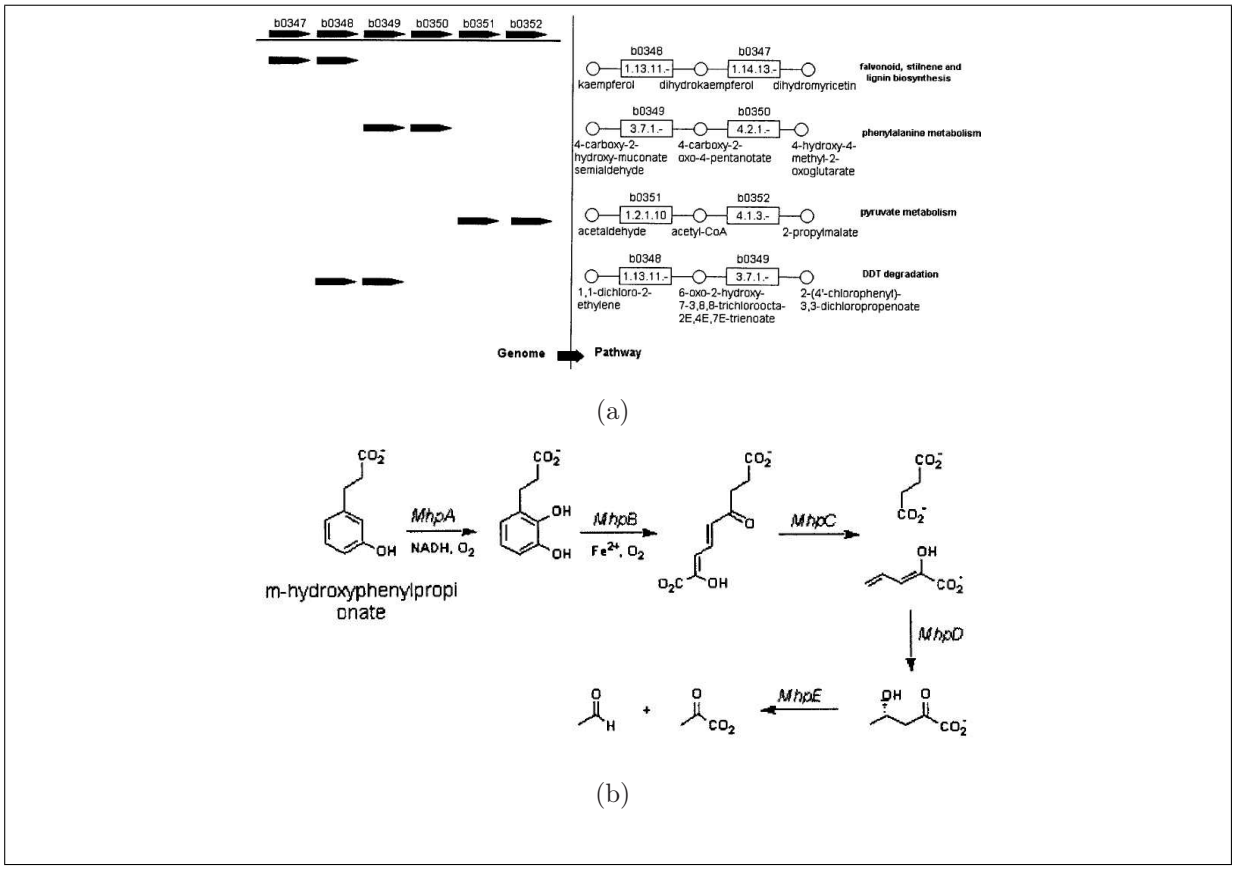


Fig. 6.10: Fusion d'opérons prédits co-orientés et successifs pour l'obtention d'un opéron de plus grande taille - La sous-figure (a) montre les opérons prédits de taille 2 qui sont fusionnés, la voie métabolique correspondante est en (b), il s'agit de la voie de dégradation du phénylpropionate

6.2.2.4 Résultats

Cette section présente les résultats de [Zheng *et al.*, 2002], obtenus à partir de 42 génomes complets (dont la majorité bactériens) avec l'approche décrite au paragraphe précédent.

La table 6.1 présente les résultats bruts obtenus.

Organisme	#opérons prédits	taille du génome	#enzymes en opérons	#enzymes détectées	$\left(\frac{\#\text{enzymes en opérons}}{\#\text{enzymes détectées}}\right)$	$\left(\frac{\#\text{enzymes en opérons}}{\#\text{opérons prédits}}\right)$	#opérons attendus
E. coli	124	4.7	374	562	0.66	3.0	22
H. influenzae	52	1.8	160	293	0.54	3.1	11
X. fastidiosa	42	2.7	148	365	0.39	3.5	5
V. cholerae	80	4.0	237	562	0.42	3.0	11
P. aeruginosa	101	6.4	290	715	0.41	2.9	14
Buchnera sp APS	30	0.7	122	224	0.54	4.1	8
P. multocida	57	2.3	169	418	0.40	3.0	12
N. meningitidis B	38	2.3	134	404	0.33	3.5	8
H. pylori	25	1.7	81	280	0.29	3.2	6
C. jejuni	33	1.7	112	335	0.33	3.4	8
R. prowazekii	25	1.1	71	182	0.39	2.8	6
M. loti	88	7.1	260	729	0.36	3.0	8
C. crescentus	48	4.1	135	372	0.36	2.8	5
B. subtilis	105	4.3	323	510	0.63	3.1	13
B. halodurans	98	4.3	308	563	0.55	3.1	13
M. genitalium	13	0.6	40	86	0.47	3.1	5
M. pneumoniae	17	0.8	50	117	0.43	2.9	6
M. pulmonis	19	1.0	60	116	0.52	3.2	4
U. urealyticum	11	0.8	33	101	0.33	3.0	3
L. lactis	62	2.4	203	367	0.55	3.3	9
S. pyogenes	46	1.9	152	283	0.54	3.3	10
S. aureus Mu50	6	2.9	21	43	0.49	3.5	0
M. tuberculosis	89	4.5	266	591	0.45	3.0	16
M. leprae	47	3.3	134	326	0.41	2.9	4
C. trachomatis	27	1.0	81	187	0.43	3.0	5
C. pneumoniae	24	1.2	75	190	0.39	3.1	4
B. burgdorferi	20	1.5	50	138	0.36	2.5	2
T. pallidum	15	1.2	44	152	0.29	2.9	5
Synechocystis	24	3.6	59	453	0.13	2.5	8
D. radiodurans	46	2.7	136	434	0.31	3.0	7
A. aeolicus	31	1.6	90	387	0.23	2.9	10
T. maritima	42	1.9	172	368	0.47	4.1	7
M. jannaschii	28	1.7	96	274	0.35	3.4	9
M. thermoautotrophicum	46	1.8	164	349	0.47	3.6	10
A. fulgidus	59	2.2	178	406	0.44	3.0	13
T. acidophilum	34	1.6	107	271	0.39	3.2	8
T. volcanium	35	1.6	116	277	0.42	3.3	7
P. horikoshii	23	1.8	80	231	0.35	3.5	4
P. abyssi	30	1.8	125	280	0.45	4.2	6
A. pernix	33	1.7	95	274	0.37	2.9	4
S. solfataricus	59	3.0	190	458	0.41	3.2	11
S. cerevisiae	33	13	74	692	0.11	2.2	16

Tab. 6.1: La liste des organismes traités dans [Zheng *et al.*, 2002] pour la recherche d'opérons à partir de voies métaboliques et les prédictions obtenues

Dans cette table, pour chaque organisme, on retrouve le nombre total d'opérons prédits, la taille du génome (en millions de paires de bases), le nombre d'enzymes en opérons, le nombre d'enzymes prédites dans le génome, la proportion qu'ont les enzymes à appartenir à un opéron et la taille moyenne des opérons. La dernière colonne indique le nombre d'opérons attendus pour le génome avec le même nombre de gènes et d'enzymes sous l'hypothèse que les enzymes sont placées au hasard sur le chromosome.

La figure 6.11 montre le nombre d'enzymes en opérons prédits en fonction du nombre

total d'enzymes prédites dans l'organisme. Les organismes avec la plus forte déviation par rapport à la moyenne sont *Escherichia coli* et *Bacillus subtilis* pour les organismes avec le plus grand rapport et *Synechocystis* et *Saccharomyces cerevisiae* pour ceux qui ont le rapport le plus faible. *Synechocystis* est une cyanobactérie, *Saccharomyces cerevisiae* est le seul organisme eucaryote de l'étude. Les génomes eucaryotes ne sont pas connus pour être organisés en opérons (même si dans le génome de certains nématodes, comme *Caenorhabditis elegans*, une fraction non négligeable des gènes appartient à des opérons [Blumenthal *et al.*, 2002]).

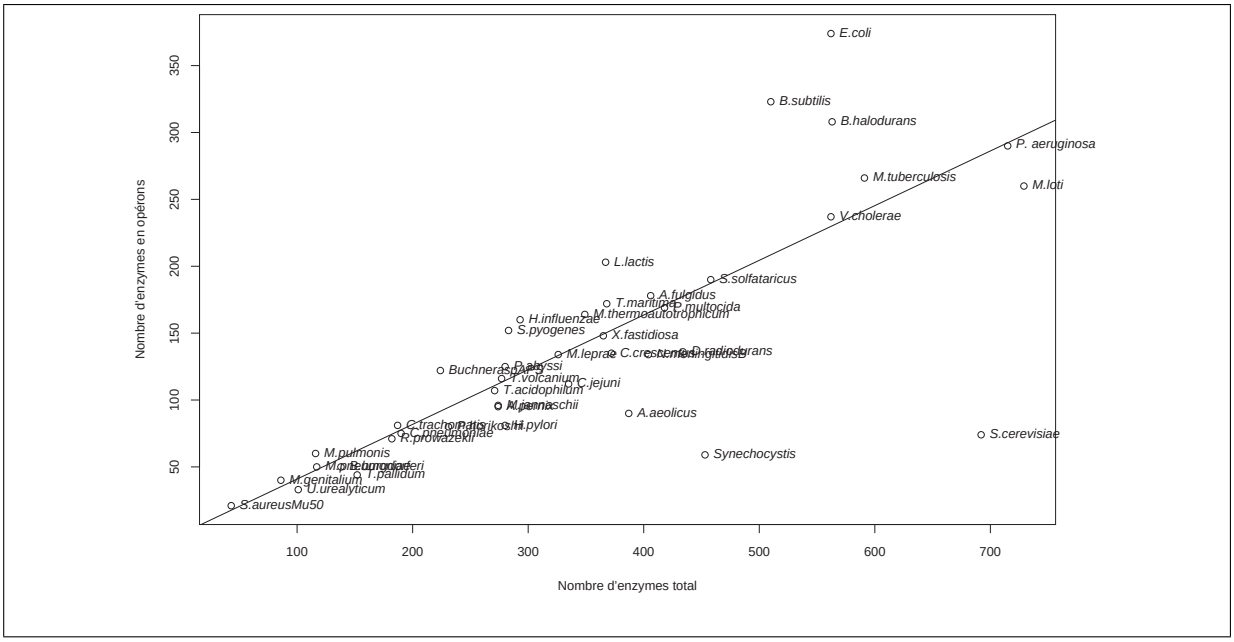


Fig. 6.11: Nombre d'enzymes en opérons prédits par [Zheng *et al.*, 2002] en fonction du nombre total d'enzymes pour 42 organismes complètement séquencés

Cette étude montre que l'organisation en opérons des gènes codant pour des enzymes impliquées dans une même voie métabolique est une caractéristique commune à quasiment tous les organismes procaryotes.

6.3 Conclusion

Dans ce chapitre, nous avons successivement étudié deux aspects différents des réseaux métaboliques : leurs caractéristiques topologiques et leur lien avec l'organisation des génomes bactériens et archæbactériens.

Au niveau topologique, on peut particulièrement retenir que certains composés sont beaucoup plus utilisés que d'autres. Un autre point à remarquer est la faible distance entre n'importe quel couple de composés. Cela semble indiquer une très grande adaptabilité

des capacités métaboliques des organismes : la transformation d'une molécule en une autre ne semblent en effet nécessiter qu'un nombre réduit d'étapes. Les caractéristiques topologiques globales des réseaux métaboliques, qui s'éloignent fortement de celles du modèle aléatoire, sont également à remarquer.

Pour ce qui est des caractéristiques génomiques des réseaux métaboliques, il est visible que l'organisation en opérons des génomes bactériens a un impact fort sur l'organisation des gènes codant pour des enzymes participant à une même voie métabolique. Toutes ces informations sur les réseaux métaboliques sont d'une grande aide lorsqu'il s'agit d'effectuer des prédictions, ces informations peuvent notamment servir à élaborer ou à perfectionner les méthodes de prédictions des voies métaboliques.

Conclusion et perspectives

Dans cette partie, nous nous sommes attachés à présenter une vue d'ensemble des travaux menés autour de la reconstruction des voies métaboliques, c'est-à-dire :

- les différentes façons de représenter et de manipuler les voies métaboliques par des graphes et des réseaux de Petri
- les deux grandes façons d'aborder le problème de la reconstruction, soit, la reconstruction par homologie et la reconstruction *ab initio*. Nous avons abordé certains problèmes spécifiques comme la prédiction des fonctions enzymatiques dans le cas de la reconstruction par homologie
- deux façons d'aborder le problème de la caractérisation des réseaux métaboliques : au niveau de leur topologie ainsi que au niveau de leur organisation sur le génome bactérien et archæbactérien

Sur la base de ces différents travaux, nous avons décidé d'attaquer le problème de la reconstruction de voies métaboliques sous deux angles différents. Le premier concerne la reconstruction *ab initio*, c'est-à-dire la recherche de réseaux et de chemins métaboliques à partir des seules données concernant les réactions et les composés (ainsi que les composés initiaux et finaux de la voie à rechercher). Le second s'inspire directement du dernier chapitre et tend à utiliser le fait que, pour les organismes procaryotes, une voie métabolique tend à être catalysée par des protéines dont les gènes associés sont co-localisés sur le chromosome.

Troisième partie

Développement de nouvelles méthodes pour la reconstruction *ab initio* de voies métaboliques

Chapitre 7

Reconstruction sous contrainte d'équilibre global

7.1 Objectif et présentation de l'approche

Les méthodes de reconstruction *ab initio* présentées au § 5.2 ont le plus souvent pour objectif de fournir l'ensemble de toutes les solutions du problème posé (soit en énumérant effectivement toutes les solutions du problème soit en fournissant une description compacte de l'espace des solutions), ce qui peut mener à un grand nombre de solutions lorsque le nombre de réactions et de composés devient important. Nous proposons donc d'appliquer des critères supplémentaires afin de restreindre la taille de cet espace de solution. Un critère naturel est de limiter la taille des solutions à la taille minimale, la taille d'une solution étant donnée par le nombre de réactions impliquées dans la solution. D'un point de vue biologique, ce choix peut être motivé par le fait que les voies métaboliques connues sont le plus souvent de taille assez restreinte.

Le formalisme des réseaux de Petri, déjà décrit au § 5.2.2, sera utilisé ici pour la description de l'approche. Le problème que nous posons est très proche du problème 3 - page 66. Il faut rappeler que dans un réseau de Petri représentant un ensemble de réactions, les réactions réversibles sont représentées par deux transitions, une pour chaque sens de la réaction.

Ce chapitre se décompose en 3 parties traitant successivement du problème, de l'algorithme mis en œuvre pour le résoudre et des expérimentations menées.

7.2 Formulation du problème

Le problème que nous traitons peut être énoncé sous la forme du problème d'optimisation suivant :

PROBLÈME **8** RECHERCHE DU PLUS PETIT SOUS-RÉSEAU INDUIT PAR UN VECTEUR DE TRANSITIONS CLOS

DONNÉES : *un réseau de Petri $R = (P, T, Pre, Post)$, trois ensembles de places $I \subseteq P$, $O \subseteq P$ et $U \subseteq P$ avec les propriétés suivantes :*

- $I \neq \emptyset$
- $O \neq \emptyset$
- *I, O et U sont tous 2 à 2 disjoints*

RÉPONSE : *le réseau de Petri $R' = (P', T', Pre', Post')$ tel qu'il existe un vecteur V de taille $|T|$ et V est un vecteur de transitions clos minimal pour R par rapport à $U \cup I \cup O$ (définition 8 - page 65), R' est le sous-réseau induit par V, et R' satisfait la contrainte :*

- *R' contient un chemin de i à o* $\quad \forall (i, o) \in I \times O$ *(définition 6 - page 64)*

MESURE : $|T'|$

OPTIMISATION : min

On notera la similarité de cette formalisation avec celle du problème 3.

Dans cette formalisation les ensembles de places I, O et U correspondent respectivement aux ensembles de composés initiaux, finaux et ubiquitaires.

Exemple :

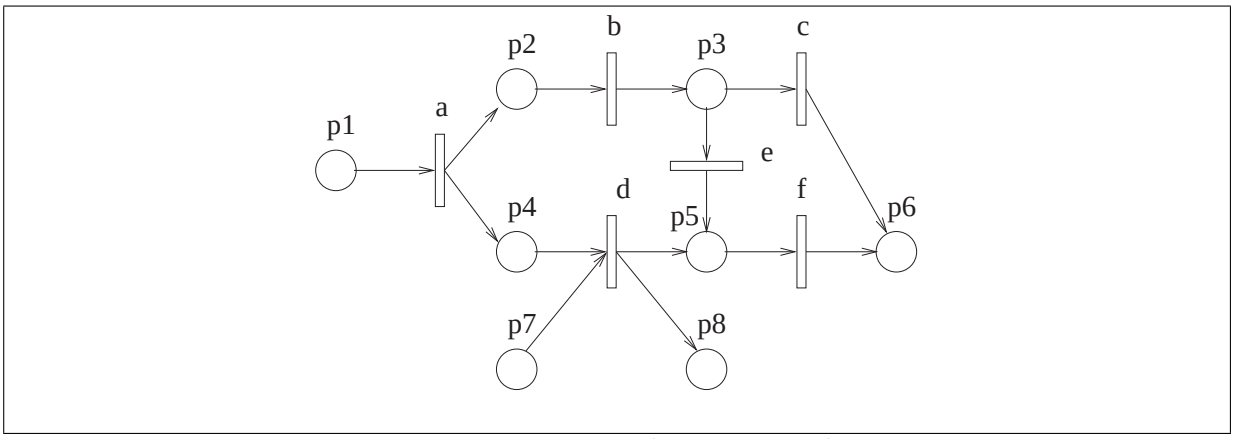


FIG. 7.1: Réseau de Petri simple

Si le problème 8 est posé avec pour données :

- le réseau de la figure 7.1
- l'ensemble de places $I = \{p1\}$
- l'ensemble de places $O = \{p6\}$

– l'ensemble de places $U = \{p7, p8\}$

Alors il y a deux réponses qui sont :

– le sous-réseau défini par l'ensemble des transitions $\{a, b, d, e, f\}$

– le sous-réseau défini par l'ensemble des transitions $\{a, b, c, d, f\}$

On constate que ces 2 ensembles de transitions permettent de produire $p6$ à partir de $p1$ en produisant et en consommant dans les mêmes quantités tous les autres composés à l'exception des composés représentés par les places $p7$ et $p8$. En conséquence, si l'ensemble des places U ne contient ni $p7$, ni $p8$ alors il n'y a aucune solution car il n'existe aucun vecteur de transitions clos pour $U \cup I \cup O$ qui produise $p6$ à partir de $p1$.

7.3 Algorithme

La stratégie que nous avons adoptée pour la résolution du problème 8 consiste à rechercher d'abord les solutions d'un problème "relaxé" dont l'ensemble des solutions inclut l'ensemble des solutions du problème initial.

Les critères utilisés pour définir cet ensemble plus grand de sous-réseaux sont des critères qualitatifs sur la structure des sous réseaux, ces critères s'appliquant, bien entendu, également aux solutions du problème initial (ces critères sont des conditions nécessaires mais non suffisantes du problème initial). Cette stratégie de résolution permet de ne pas se focaliser directement sur les solutions numériques du problème (l'assignation de valeurs aux variables du système d'inéquations sous-jacent à notre problème) et permet ainsi de réduire la complexité du problème. Si un sous-réseau est solution du problème relaxé, il faut tester si c'est une solution du problème initial, mais dans le cas contraire, ce test n'est pas nécessaire.

Dans un premier temps, le problème relaxé est explicitement énoncé. Ensuite, l'espace de recherche de ce problème est présenté. La manière selon laquelle est exploré l'espace de recherche est importante et peut amener à une réduction drastique des calculs. Les heuristiques de parcours que nous avons testées sont présentées ainsi que leurs performances. Ensuite, nous décrivons comment sont évaluées les solutions du problème relaxé pour savoir si elles sont également solutions du problème initial.

7.3.1 Définition du problème "relaxé"

Soit le sous-réseau $R' = (P', T', Pre', Post')$ solution du problème 8 avec pour paramètres le réseau de Petri $R = (P, T, Pre, Post)$ et les ensembles I,O et U (qui respectent les contraintes énoncées dans la définition du problème), alors R' respecte forcément les propriétés suivantes :

- $p \in P' \setminus \{I \cup O \cup U\}, \exists (t_1, t_2) \in T' \times T' / Post'(p, t_1) > 0$ et $Pre'(p, t_2) < 0$ (pour chaque place intermédiaire p, T' doit contenir une transition consommant la ressource associée à la place p et une transition produisant la ressource associée à la place p)
- R' contient un chemin de i à o $\forall i \in I, \forall o \in O$ (définition 6 - page 64)

Le problème relaxé est donc défini de la façon suivante :

Problème **9** Recherche de sous-réseaux de Petri contraints
Données : *un réseau de Petri $R = (P, T, Pre, Post)$, trois ensembles de places $I \subseteq P$, $O \subseteq P$ et $U \subseteq P$ avec les propriétés suivantes :*

- $I \neq \emptyset$
- $O \neq \emptyset$
- *I, O et U sont tous 2 à 2 disjoints*

Réponse : *les sous-réseaux de Petri $R' = (P', T', Pre', Post')$ de R tels que les R' satisfont les contraintes :*

- $\exists (t_1, t_2) \in T' \times T' / Post(p, t_1) > 0$ *et* $Pre(p, t_2) < 0$ $\forall p \in P' \setminus \{I \cup O \cup U\}$
- *R' contient un chemin de i à o* $\forall (i, o) \in I \times O$ *(définition 6 - page 64)*

Exemple : si le problème 9 est posé avec pour données :

- le réseau de la figure 7.1
- l'ensemble de places $I = \{p1\}$
- l'ensemble de places $O = \{p6\}$
- l'ensemble de places $U = \{p7, p8\}$

alors le réseau entier est une des solutions du problème relaxé (en plus des deux sous-réseaux $\{a, b, d, e, f\}$ et $\{a, b, c, d, f\}$) car :

- pour chaque place différente de $\{p1, p6, p7, p8\}$ ce réseau contient une transition qui consomme cette place et une transition qui la produit
- il y a un chemin de $p1$ à $p6$

7.3.2 Espace de recherche et stratégie d'énumération

Dans ce paragraphe sont décrits l'espace de recherche commun aux deux problèmes 8 et 9 ainsi que les stratégies d'énumération communes mises en œuvre pour son parcours.

7.3.2.1 Description de l'espace de recherche

Comme nous recherchons un sous-réseau du réseau de Petri initial $R = (P, T, Pre, Post)$, l'espace de recherche implicitement parcouru est composé de tous les sous-réseaux de Petri du réseau de Petri initial. Un sous-réseau est défini par un sous-ensemble des transitions

du réseau de Petri initial, il en a donc $2^{|T|}$ différents qui peuvent être ordonnés par la relation d'inclusion d'ensemble $\subset$ dans un treillis (comme sur la figure 7.2).

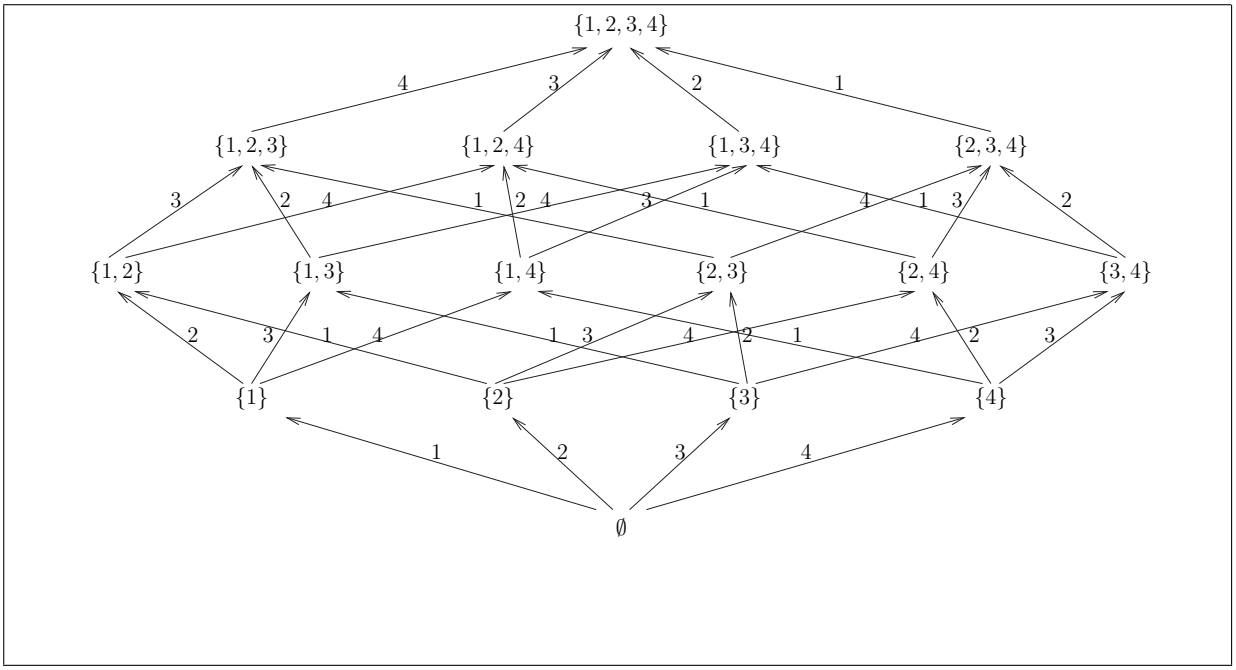


Fig. 7.2: Treillis défini par l'ensemble des parties de $\{1, 2, 3, 4\}$ et la relation $\subset$

Pour les deux problèmes (initial et relaxé) l'espace de recherche est le même, cependant, les ensembles de solutions ne sont pas les mêmes. L'ensemble des solutions du problème relaxé inclut l'ensemble des solutions du problème initial. Comme la taille d'une solution est donnée par le nombre de transitions du sous-réseau de Petri, toutes les solutions du problème initial sont au même niveau dans le treillis. La taille de la plus petite solution du problème relaxé est forcément inférieure ou égale à la taille des solutions du problème initial.

7.3.2.2 Stratégies de parcours de l'espace de recherche

Pour la recherche des solutions du problème relaxé, il semble judicieux de commencer l'exploration à la base du treillis, qui représente le sous-réseau vide, et de le parcourir en ajoutant successivement des transitions, ce qui fait remonter dans le treillis et augmente le nombre de transitions associées au sous-réseau exploré. En effet, les solutions aux problèmes 8 et 9 sont généralement petites par rapport à l'élément maximum du treillis.

L'efficacité de la recherche des solutions dans l'espace de recherche peut être améliorée de plusieurs façons qui sont décrites dans les paragraphes suivants.

Réduction de l'espace de recherche La propriété selon laquelle toutes les solutions doivent contenir les chemins entre tous les couples $(substrat, produit)$ implique que les sous-réseaux solutions sont connexes. Cette propriété permet de restreindre la recherche aux sous-réseaux dont chaque composante connexe est reliée à une place de l'ensemble I (puisque c'est une condition nécessaire). Cela limite beaucoup les possibilités de choix de la transition à ajouter pour passer d'un nœud à un autre dans le treillis des sous-réseaux. De plus on peut également imposer que l'ajout d'une transition soit possible uniquement si une des places qui correspond à une ressource consommée par cette transition est d'ores et déjà présente dans le sous-réseau (au départ, cet ensemble est $I \cup U$).

Exemple : la figure 7.3 montre un réseau ayant 6 transitions et l'espace de recherche réduit associé au problème 9 si les données du problème sont $I = \{p1\}$, $O = \{p6\}$ et $U = \emptyset$ et que les règles énoncées au dessus sont imposées. Ces règles permettent dans ce cas de restreindre l'espace de recherche de $2^6 = 64$ à 19 sous-réseaux. A titre d'illustration, on remarque que le sous-réseau composé uniquement de la transition b (qui devrait se trouver à la base du treillis) n'est pas exploré car le sous-réseau correspondant n'est pas connexe (en partant de $I \cup U$. Notons que cela n'empêche pas, plus tard, de retrouver des sous-réseaux qui contiennent la transition b.

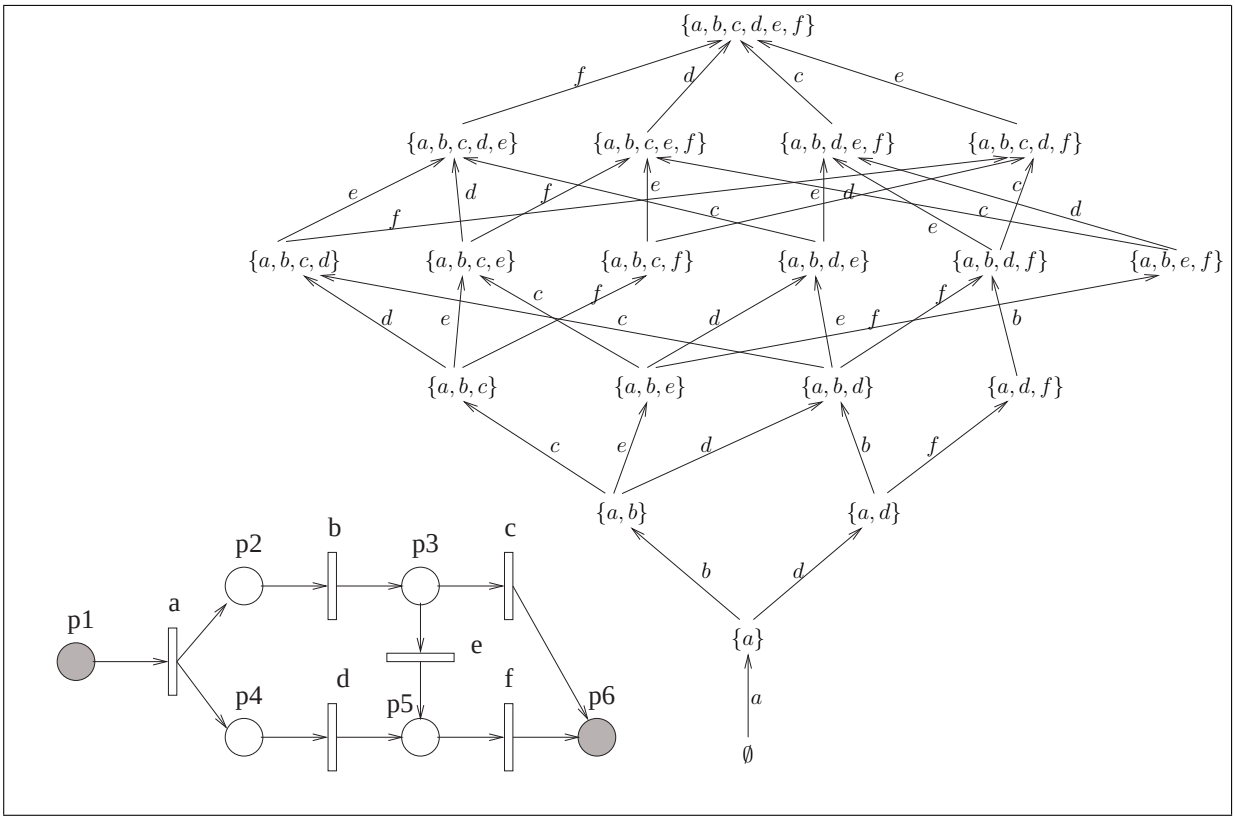


Fig. 7.3: Réduction de l'espace de recherche en limitant l'exploration aux sous-réseaux connexes

Fonction d'évaluation et fonctions d'estimation Comme énoncé précédemment, la taille des solutions aux deux problèmes est donnée par le nombre de transitions formant le sous-réseau.

Un nœud du treillis ne représentant pas une solution du problème relaxé peut avoir parmi ses successeurs (les éléments qui lui sont supérieurs dans le treillis) une solution de ce problème. L'évaluation d'un nœud non solution est basée sur la distance qui le sépare d'un nœud solution dans le treillis. Plus cette distance est faible, plus l'évaluation de ce nœud doit être bonne. L'évaluation d'un nœud non solution peut ainsi être définie suivant une somme de deux termes. Le premier terme de la somme est la taille effective du nœud, c'est-à-dire la distance séparant le nœud de la base du treillis ($d(n)$). Le deuxième terme est la longueur du chemin entre le nœud et le nœud solution le plus proche ($h(n)$). Ceci est illustré sur la figure 7.4. L'évaluation d'un nœud vaut donc $e(n) = d(n) + h(n)$.

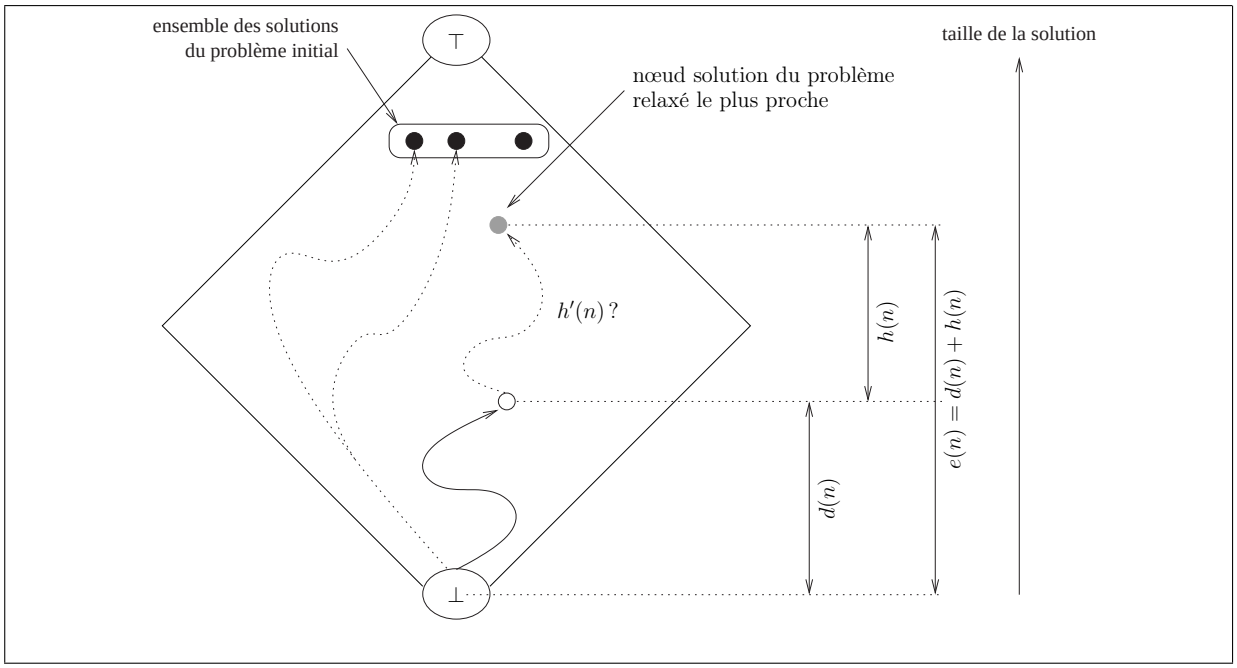


Fig. 7.4: Fonction d'évaluation d'un nœud non solution dans l'espace de recherche

Le problème est que la valeur $h(n)$ est le plus souvent inconnue, il faut donc recourir à des fonctions d'estimation notées $h'(n)$ qui estiment la valeur de $h(n)$. Dans le cas présent, une fonction d'estimation a pour objectif de fournir une borne inférieure sur le nombre de transitions à parcourir dans le treillis pour aller d'un nœud n non solution au nœud solution le plus proche dans le treillis. L'estimation triviale consiste à poser $h'(n) = 1$ ce qui signifie que le nœud courant n n'est pas solution et qu'il faut franchir au moins une transition dans le treillis pour arriver à un nœud solution. Il est souhaitable d'améliorer cette estimation triviale car elle permet d'améliorer le parcours de l'espace de recherche.

Etant données les restrictions imposées à l'espace de recherche, un nœud non solution

représente un sous-réseau qui présente la propriété particulière que certaines des places de l'ensemble O peuvent ne pas avoir été atteintes (il n'y a aucune transition transition dans le sous-réseau de Petri qui produise cette place).

Une première fonction $h'_1(n)$ estimant le nombre de transitions à ajouter au sous-réseau courant consiste à estimer le nombre minimum de transitions à ajouter au sous-réseau pour qu'au moins une des places de l'ensemble O soit atteinte à partir d'une place de I. Comme illustré sur la figure 7.5, toutes les places ne peuvent pas servir pour ajouter des transitions dans ce cas précis. Sont exclues les places de O, celles de U, ainsi que les places faisant partie du sous-réseau qui n'ont pas de transitions entrantes (voir la figure 7.5).

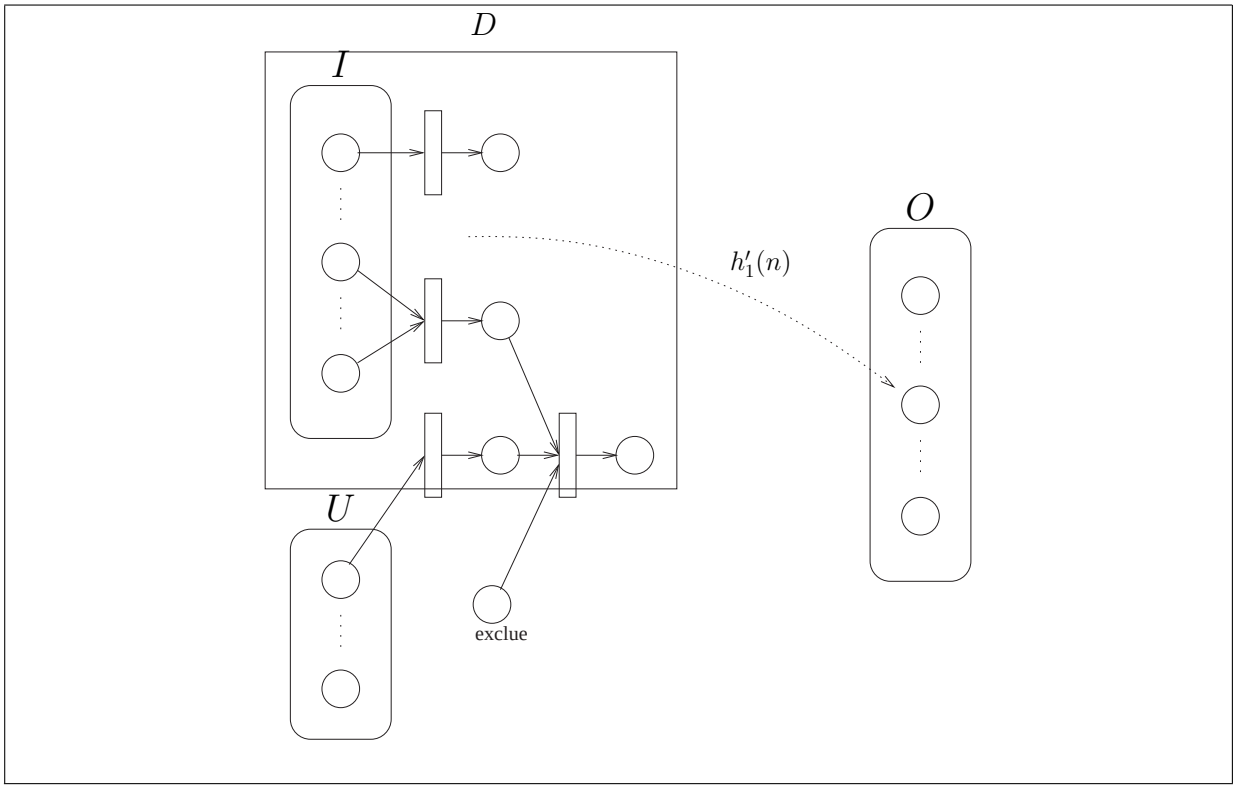


Fig. 7.5: Fonction d'estimation $h'_1(n)$ - $h'_1(n)$ a pour but d'estimer le nombre minimum de transitions à ajouter pour qu'il existe un chemin d'une place de I à une place de O

Si l'ensemble D représente l'ensemble des places du sous-réseau qui ont une transition entrante plus l'ensemble I, la fonction $h'_1(n)$ peut être définie de la façon suivante :

$$h'_1(n) = \min \left(\underset{(x,y)\in(D\times O)}{\text{distanceMin}} (x, y) \right)$$

Une autre fonction d'estimation $h'_2(n)$ consiste à estimer le nombre minimum de transitions qu'il faut rajouter au sous-réseau pour que toutes ses places aient au moins une transition entrante (sauf pour les places de I et de U), et une transition sortante (sauf

pour les places de O et de U). Ce problème peut être modélisé sous la forme d'un problème connu appelé Couverture d'Ensemble de Taille Minimale défini ci-dessous et illustré sur la figure 7.6.

Problème **10** Couverture d'Ensemble de Taille Minimale
Données : *une collection* $C = \{C_1, \ldots, C_n\}$ *de sous-ensembles d'un ensemble fini* S
Réponse : *un sous-ensemble* $C' \subseteq C$ *tel que* $\bigcup_{C_i \in C'} C_i = S$
Mesure : $|C'|$
Optimisation : min

Exemple : la figure 7.6 donne un exemple d'instance du problème Couverture d'Ensemble de Taille Minimale et de sa solution optimale.

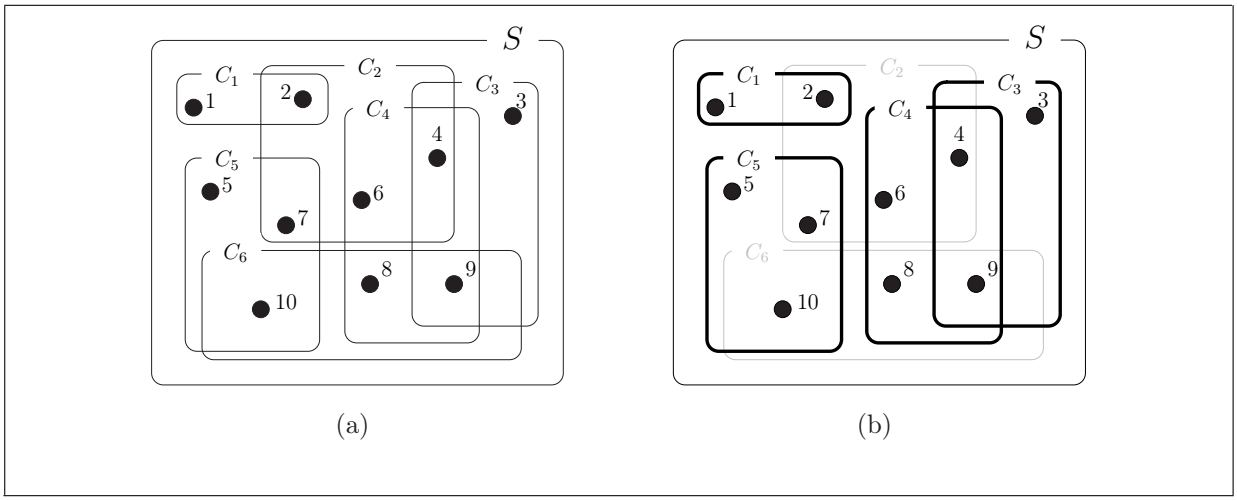


Fig. 7.6: Illustration du problème Couverture d'Ensemble de Taille Minimale - (a) un exemple d'instance et (b) sa solution optimale

Dans notre cas, l'ensemble S est l'union de deux ensembles de places D^- et D^+ qui sont respectivement l'ensemble des places n'ayant aucune transition sortante (exceptées les places de $I \cup U$) et l'ensemble des places n'ayant aucune transition entrante (exceptées les places de $O \cup U$) comme illustré sur la figure 7.7. La collection C est définie par les transitions que l'on peut ajouter au sous-réseau. A chaque transition t pouvant être ajoutée correspond un C_i. Le C_i associé à la transition t couvre une place p de S si et seulement si :

– $p \in D^-$ et t consomme la ressource associée à la place p

ou

– $p \in D^+$ et t produit la ressource associée à la place p

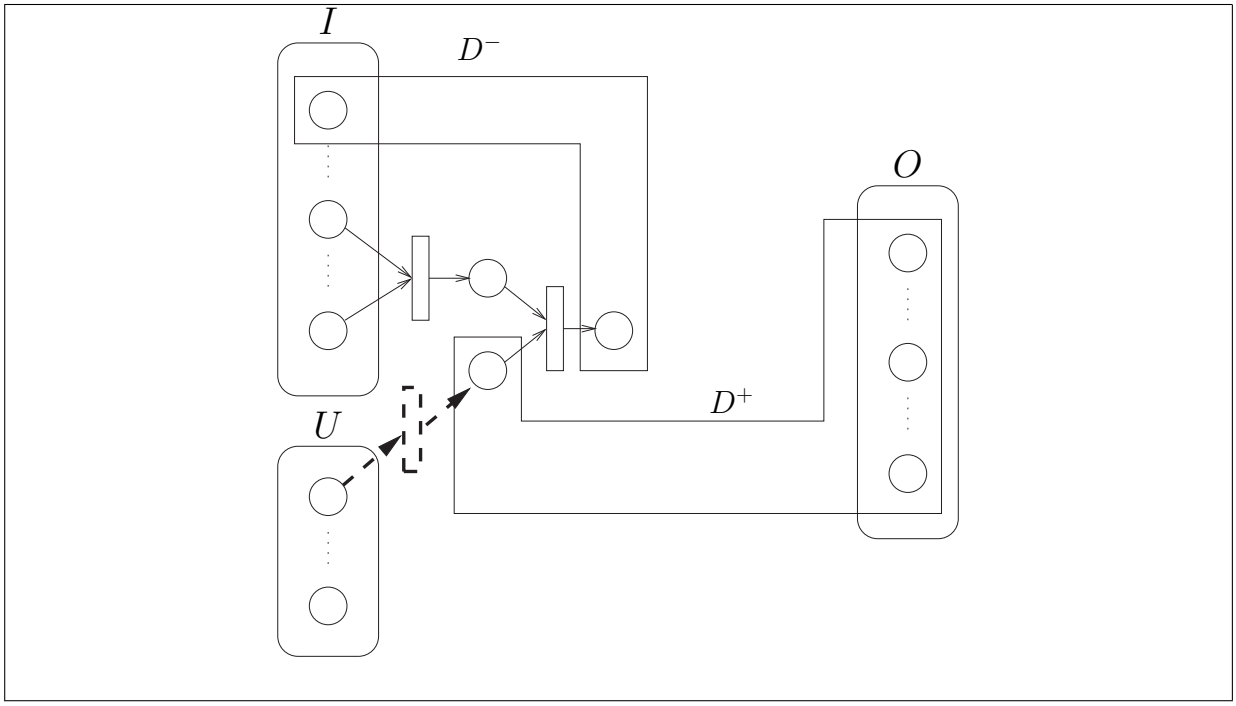


FIG. 7.7: Fonction d'estimation $h'_2(n)$ - Couverture minimale des places ouvertes par des transitions, illustration des ensembles D^+ et D^-. La transition dessinée en traits pointillées permet de couvrir une des places de l'ensemble D^+

Comme le problème COUVERTURE D'ENSEMBLE DE TAILLE MINIMALE est un problème $\mathcal{NP}$-difficile [Crescenzi and Kann, 1998], il n'est pas judicieux de le résoudre exactement et il vaut mieux utiliser un algorithme rendant une solution approchée. La fonction d'estimation $h'_2(n)$ doit fournir une borne inférieure et non une borne supérieure à la solution du problème. Un algorithme possible consiste en un algorithme glouton qui marque successivement les éléments s_i de S ainsi que les autres éléments de S appartenant aux mêmes C_k que s_i. L'algorithme se termine lorsque tous les s_i ont été marqués. Le nombre de marquages nécessaires au cours de l'exécution de l'algorithme donne une solution obligatoirement inférieure à la solution optimale du problème. L'algorithme est donné en pseudo-code (voir algorithme 7.1).

Exemple : pour l'instance de la figure 7.6, l'approximation rendue par l'algorithme glouton est de 4.

Il est également possible de combiner les deux estimations $h'_2(n)$ et $h'_1(n)$ pour n'en faire qu'une seule. Il faut alors faire attention à ne pas compter plusieurs fois les mêmes transitions. En conséquence, il faut retrancher à l'estimation de la distance h'_1 une longueur de 2 qui correspond à la longueur représentée par les transitions qui sont utilisées pour couvrir les places lors de la seconde estimation. L'expression pour le calcul de l'estimation est alors $h'(n) = h'_2(n) + max(0, h'_1(n) - 2)$.

```
Fonction GloutonCouvertureMin → Entier
Paramètre : Ensemble : S,
            Ensemble-d-ensembles : C ;
Variable : Tableau-de-booléens : M,
           Booléen : m,
           Entier : compteur ;
début
    compteur ← 0;
    pour chaque C_k ∈ C faire
        M[C_k] ← FAUX;
    pour chaque s_i ∈ S faire
        m ← FAUX;
        pour chaque C_k ∈ C tel que s_i ∈ C_k faire
            m ← m ∨ M[C_k];
        si ¬m alors
            /*
            on effectue le marquage de s_i
            ainsi que tous les éléments de S qui
            appartiennent à un même C_k que s_i
            */ compteur ← compteur + 1;
            pour chaque C_k ∈ C tel que s_i ∈ C_k faire
                M[C_k] ← VRAI;
    → compteur;
fin
```

ALG. **7.1:** Algorithme glouton donnant une borne inférieure pour le problème COUVERTURE D'ENSEMBLE DE TAILLE MINIMALE

Méthode d'énumération A* Plusieurs stratégies d'énumération sont possibles afin de parcourir les nœuds de l'espace de recherche notamment le parcours en largeur d'abord. Cependant la disponibilité de fonctions d'estimations permet d'utiliser une variante de cette stratégie de parcours de l'espace de recherche : la méthode A*.

En effet, la connaissance d'une estimation de la distance séparant un nœud avec le nœud solution le plus proche permet de donner la priorité aux nœuds sensés mener plus rapidement à une solution.

Avantages et inconvénients de A* Si la fonction d'estimation est optimiste (*i.e.* qu'elle donne une valeur inférieure ou égale à la distance réelle séparant le nœud courant du nœud solution le plus proche), alors A* garantit que la première solution trouvée est la plus petite. Dans le cas où la fonction d'estimation donne la distance exacte séparant un nœud de son nœud solution le plus proche, A* trouve directement la solution. Dans le cas où la fonction d'estimation n'estime rien et renvoie toujours la valeur 1, alors A* est semblable au parcours en largeur d'abord.

Comme le parcours en largeur d'abord, A* doit conserver tous les nœuds en attente d'exploration qui représente la frontière entre les nœuds déjà traités et les nœuds non encore explorés. Cette frontière, aussi appelée liste "ouverte", peut donc croître très rapidement.

On notera enfin que A* est facilement adaptable si l'on souhaite obtenir toutes les solutions du problème initial d'une taille inférieure à un seuil donné en paramètre. Il suffit de continuer l'énumération après avoir trouvé la première solution, l'énumération doit s'arrêter lorsque la taille des nœuds testés est trop grande.

7.3.2.3 Evaluation des solutions du problème relaxé pour le problème initial

Si un nœud de l'espace de recherche est solution du problème relaxé, il faut vérifier si il est également solution du problème initial. Cette vérification consiste à établir si il existe un vecteur clos minimal pour le réseau initial R par rapport à $U \cup I \cup O$ induisant ce sous-réseau (définitions 8 et 9 - page 65). Ce problème peut être résolu efficacement par des approches voisines de celles présentées au § 5.2.1.

7.4 Mise en œuvre et expérimentations

Un programme suivant les points énoncés au paragraphe précédent a été implémenté en langage *C*.

7.4.1 Données

Pour tous les tests effectués le jeu de données est composé des réactions disponibles dans la banque LIGAND/KEGG [Goto *et al.*, 2002] d'avril 2003. 5422 réactions ont été extraites de la banque impliquant 4599 composés. Comme les réactions ont toutes été considérées comme réversibles, le réseau de Petri associé compte 10844 transitions et 4599 places.

7.4.2 Application à la glycolyse

Pour reconstruire la glycolyse, les ensembles de substrats passés en paramètres sont $I = \{\alpha\text{-D-glucose 6-phosphate}\}$, $O = \{\text{pyruvate}\}$ et l'ensemble des composés U pour lequel la contrainte d'équilibre est relaxée est $U = \{$eau, ATP, NAD^+, NADH, NADPH, $NADP^+$, O_2, ADP, orthophosphate, coenzyme A, CO_2, pyrophosphate, NH_3, H_2O_2, accepteur, accepteur réduit, H+, $(\text{phosphate})^n\}$.

7.4.2.1 Performances de l'algorithme

Résultats bruts Le tableau 7.1 montre le nombre de solutions au problème initial pour la glycolyse et le temps de calcul pour différentes tailles maximales des solutions. Tous les résultats présentés dans les tableaux suivants ont été obtenus sur une machine PIII 1GHz avec 1Go de mémoire. "Echec" signifie que la quantité de mémoire nécessaire a l'exécution du programme est excessive (*i.e.* l'espace mémoire nécessaire (taille de la liste "ouverte") est trop important ($>$ 1Go)).

Substrat	Produit	Taille des solutions	Nombre de solutions	Temps de calcul
α-D-glucose 6P	pyruvate	6 (min)	8	1 min 14 s
α-D-glucose 6P	pyruvate	7	62	10 min 37 s
α-D-glucose 6P	pyruvate	8	**Echec**	

Tab. 7.1: Résumés des expérimentations pour la glycolyse avec estimation "distance et couverture"

Influence de la fonction d'estimation Le tableau 7.2 montre l'influence de la fonction d'estimation sur la fraction de l'espace explorée. Sans utiliser la fonction d'estimation certains calculs ne peuvent pas être effectués. Le rapport entre les versions avec et sans fonction d'estimation est dans ce cas marqué par un '?'. Cependant, même avec une fraction très réduite de sous-réseaux testés, la combinatoire est telle qu'il nous a été impossible de produire des solutions d'une taille supérieure à 7 étapes.

Sans estimation	Taille des solutions	Nombre de nœuds testés	
	4	9552	
	5	227600	
	6	**Echec**	
Couverture	Taille des solutions	Nombre de nœuds testés	Rapport
	4	191	0,0199
	5	1717	0,0075
	6	16817	?
	7	169997	?
	8	**Echec**	
Distance et Couverture	Taille des solutions	Nombre de nœuds testés	Rapport
	4	93	0,0097
	5	1115	0,0048
	6	12278	?
	7	138339	?
	8	**Echec**	

Tab. 7.2: Influence de la fonction d'estimation

7.4.2.2 Discussion

La taille maximale jusqu'à laquelle l'algorithme reste efficace (en temps de calcul et surtout en ce qui concerne l'espace mémoire requis) est relativement faible et ne permet la reconstruction que de réseaux relativement petits. Il est impossible de reconstituer la voie habituelle de la glycolyse grâce à cette approche car cette voie compte 9 réactions (à partir du α-D-glucose 6 phosphate pour aller jusqu'au pyruvate), cependant, des solutions ont tout de même été trouvées.

7.4.3 Application à la biosynthèse du tryptophane

Le second test effectué consistait à rechercher les réseaux consommant du chorismate pour produire du tryptophane ($I = \{\text{chorismate}\}$, $O = \{\text{tryptophane}\}$ et $U = \{$eau, ATP, NAD^+, NADH, NADPH, $NADP^+$, O_2, ADP, orthophosphate, coenzyme A, CO_2, pyrophosphate, NH_3, H_2O_2, accepteur, accepteur réduit, H+, $(\text{phosphate})^n\}$ (liste inchangée par rapport à l'application précédente)).

7.4.3.1 Performances de l'algorithme et résultats bruts

Les résultats obtenus sont résumés dans le tableau 7.3.

Substrat	Produit	Taille des solutions	Nombre de solutions	Nombre de nœuds testés	Temps de calcul
chorismate	tryptophane	5 (min)	1	1073	16 s
chorismate	tryptophane	6	4	14341	1 min 4 s
chorismate	tryptophane	7	21	190784	10 min 41 s
chorismate	tryptophane	8	110	1563805	1 h 46 min 55 s
chorismate	tryptophane	9		**Echec**	

Tab. 7.3: Résumés des expérimentations pour la biosynthèse du tryptophane avec estimation "distance et couverture"

7.4.3.2 Discussion

La figure 7.8 montre les 4 réseaux solutions de taille 6 ou moins.

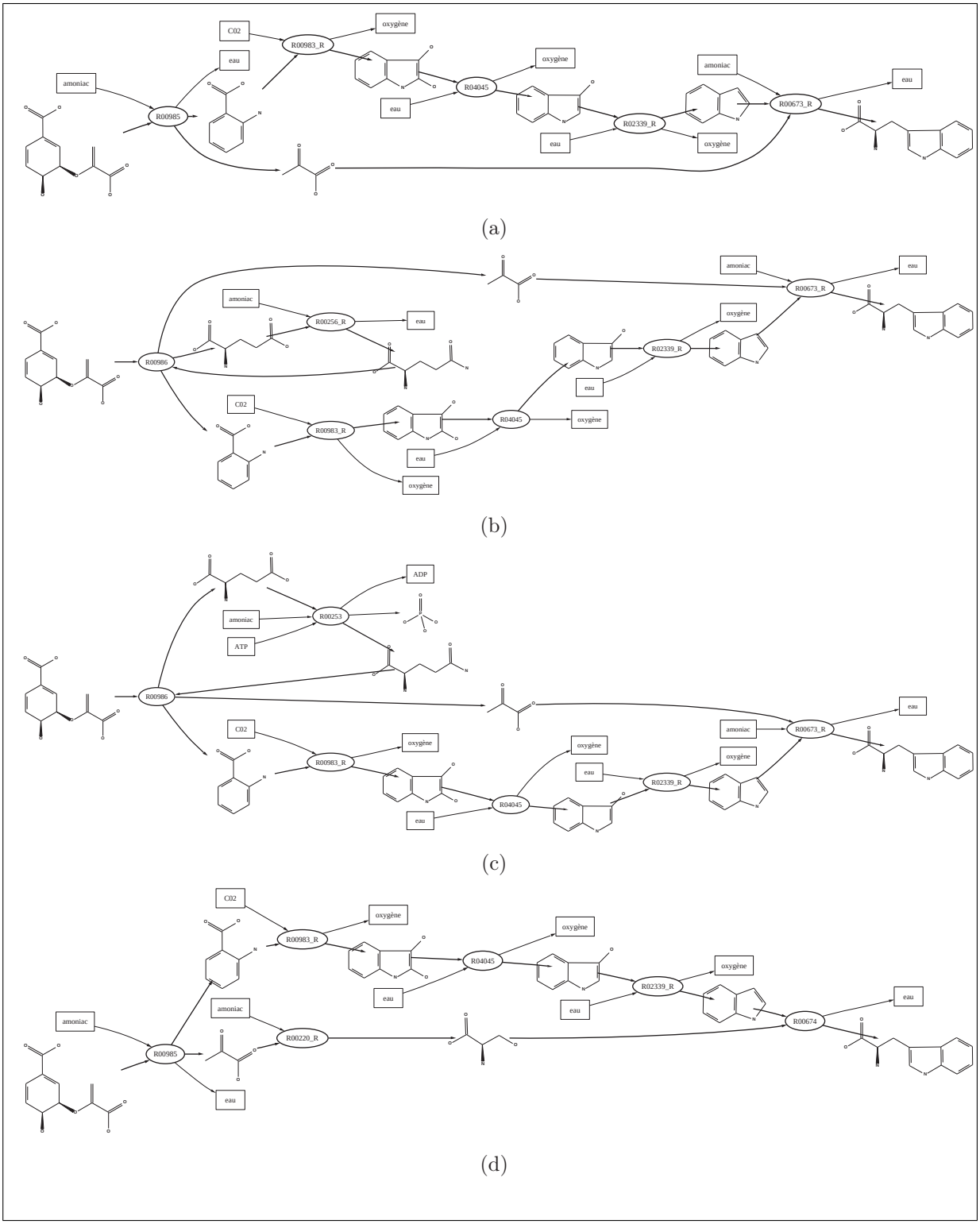


Fig. 7.8: Les 4 réseaux différents trouvés de taille 5 ou 6 pour la biosynthèse du tryptophane

Bien que toutes les réactions qui composent la véritable voie de biosynthèse du tryptophane, illustrées sur la figure 7.9 (5 étapes), soient contenues dans la base, cette voie ne fait pas partie des 4 solutions trouvées.

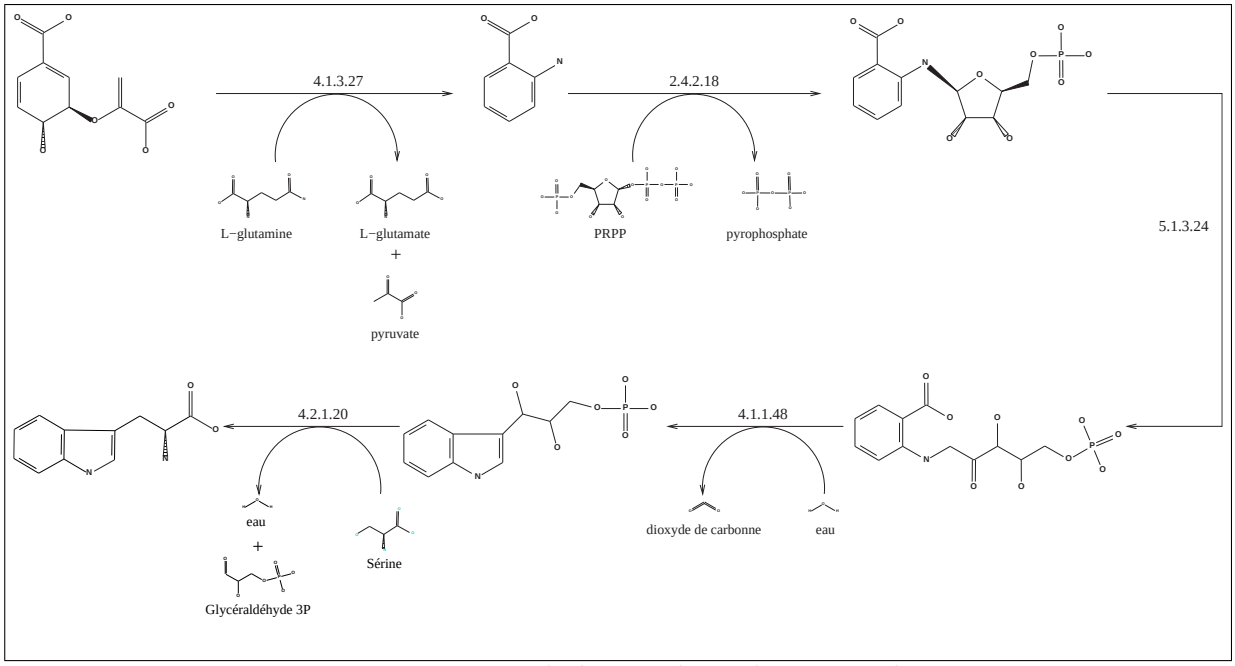


Fig. 7.9: La voie de biosynthèse du tryptophane

L'observation de la voie biosynthèse du tryptophane permet de comprendre pourquoi cette voie n'a pas pu être trouvée.

Cela ne provient pas de l'algorithme mais du choix des composés "secondaires" des réactions (définis dans l'ensemble U). Parmi les 5 réactions impliquées dans cette voie, seule la troisième étape ne fait pas intervenir de composés secondaires. Pour les quatre autres étapes, parmi les composés secondaires, seulement trois sur les neuf (eau, dioxyde de carbone et pyrophosphate) font partie de l'ensemble des composés pour lesquels la contrainte d'équilibre a été relâchée. Ainsi, il a été impossible de retrouver la voie de biosynthèse telle qu'on la connaît.

Pour certains des composés comme la glutamine ou le glutamate (utilisés dans la première réaction), leur présence ou non dans l'ensemble des composés à ne pas équilibrer peut se discuter. En effet, pour le cas particulier de la glutamine et du glutamate, ils sont souvent utilisés en couple pour permettre l'échange d'un groupement aminé (comme c'est la cas dans la voie de biosynthèse du tryptophane). Par contre pour des composés comme le PRPP ou la sérine, il est difficile de justifier leur présence dans l'ensemble des composés à ne pas équilibrer, à moins que des connaissances particulières sur la voie à reconstruire justifie ce choix.

7.5 Conclusion

L'observation des résultats obtenus sur la glycolyse et la biosynthèse du tryptophane amènent à plusieurs commentaires sur cette première approche :

- la taille des réseaux constructibles par la méthode est faible : au delà d'une taille de 8 ou 9 réactions, la reconstruction devient impossible car elle nécessite un espace mémoire trop important, et ce malgré les efforts entrepris pour limiter la partie de l'espace de recherche effectivement explorée.
- l'inclusion ou non de certains composés dans l'ensemble des composés pour lesquels la contrainte d'équilibre est relaxée a une incidence importante sur les réseaux trouvés par l'algorithme. De plus, cet ensemble paraît difficile à construire de manière universelle car il peut éventuellement varier suivant les voies recherchées.

Les objectifs fixés en introduction de ce chapitre ne sont donc pas remplis par l'approche développée :

- les temps de calcul et la mémoire requise sont trop élevés[4], et surtout
- les résultats rendus sont peu conformes aux attentes

La contrainte d'équilibre paraît être trop restrictive et influence trop les réseaux rendus. Bien que ne remplissant pas les objectifs fixés le développement de cette approche a donc permis de comprendre pourquoi la contrainte d'équilibre sur les composés intermédiaires n'est pas un bon critère dans le cas des applications testées.

L'approche présentée dans le chapitre suivant tient compte de ces observations.

[4]Afin d'éviter le problème de l'explosion de la quantité de mémoire requise, il est possible d'explorer l'espace de recherche en profondeur d'abord. Cette solution n'est cependant pas satisfaisante car les temps de calcul rendent alors l'utilisation du programme impossible sur des jeux de données réels

Chapitre 8

Reconstruction par recherche de flux d'atomes maximaux

Dans ce chapitre, une seconde façon d'aborder le problème de la reconstruction *ab initio* est abordée. L'objectif et la présentation de l'approche puis sa décomposition en deux sous-problèmes indépendants sont d'abord présentés. Chacun des deux sous-problèmes est ensuite décrit en détail avec la méthode de résolution employée pour le résoudre. Enfin, un certain nombre d'expérimentations sont présentées.

8.1 Objectif et présentation de l'approche

Par rapport à l'approche précédente, cette seconde approche adopte un point de vue radicalement différent. Contrairement à l'approche précédente, ce sont des chemins entre deux composés qui vont être maintenant recherchés et non plus des réseaux. Afin de limiter le nombre de chemins possibles entre deux composés, nous nous appuyons sur une modélisation du problème proche de celle présentée au § 5.4 qui tient compte de la structure chimique des composés. Cependant, plutôt que de rechercher les chemins que peuvent suivre chaque atome d'un composé initial vers un composé final, comme dans l'approche présentée au § 5.4, nous restreignons la recherche aux chemins correspondant à des successions de réactions qui garantissent que le nombre d'atomes transférés entre les deux composés est maximum (ou au dessus d'un seuil fixé).

L'hypothèse sous jacente à cette approche est que les réactions successives d'une voie métabolique ont tendance à minimiser le nombre de réarrangements atomiques effectués sur les composés pour obtenir le composé final. En conséquence, une grande partie des atomes constituant le composé initial devrait être transférée au composé final. A titre d'exemple, la figure 8.1 montre les atomes transférés par une des voies de biosynthèse de

la méthionine.

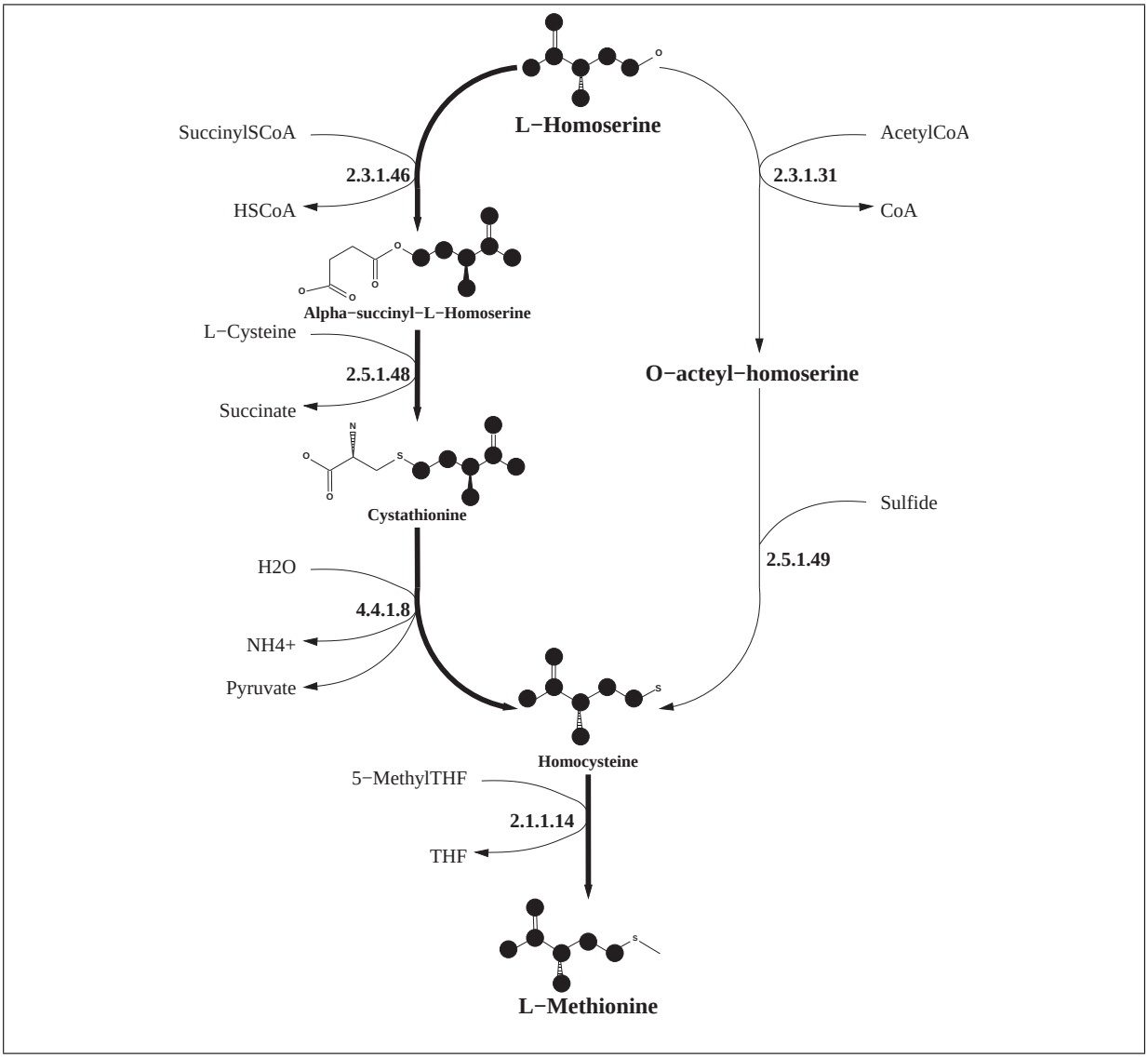


Fig. 8.1: Les atomes transférés par une des voies de biosynthèse de la méthionine - les sept atomes marqués sont successivement transférés par les réactions. Ce marquage des atomes transférés permet de mettre en évidence la fonction de cette voie qui est l'adjonction d'un atome de soufre et de carbone à la place d'un atome d'oxygène à la L-Homosérine pour donner la L-Méthionine

8.2 Décomposition du problème

Afin de calculer, pour une succession de réactions, quels atomes sont transférés entre deux composés, il faut, pour chaque réaction, connaître la correspondance entre les atomes des composés impliqués (comme dans l'approche présentée au § 5.4). Ces correspondances

ne sont pas fournies dans les bases de données de réactions. En conséquence, il faut soit traiter manuellement les réactions, soit, de préférence, les traiter automatiquement pour obtenir les correspondances atomiques qu'elles induisent. Une fois obtenues, ces correspondances peuvent être stockées pour toute utilisation future.

Une fois ces correspondances atomiques connues, il est possible d'extraire, pour chaque couple de composés $(substrat, produit)$ dans chaque réaction, une fonction injective partielle entre les deux ensembles d'atomes des composés. Cette injection partielle décrit le transfert atomique induit par la réaction entre ces deux composés (comme illustré sur la figure 8.2).

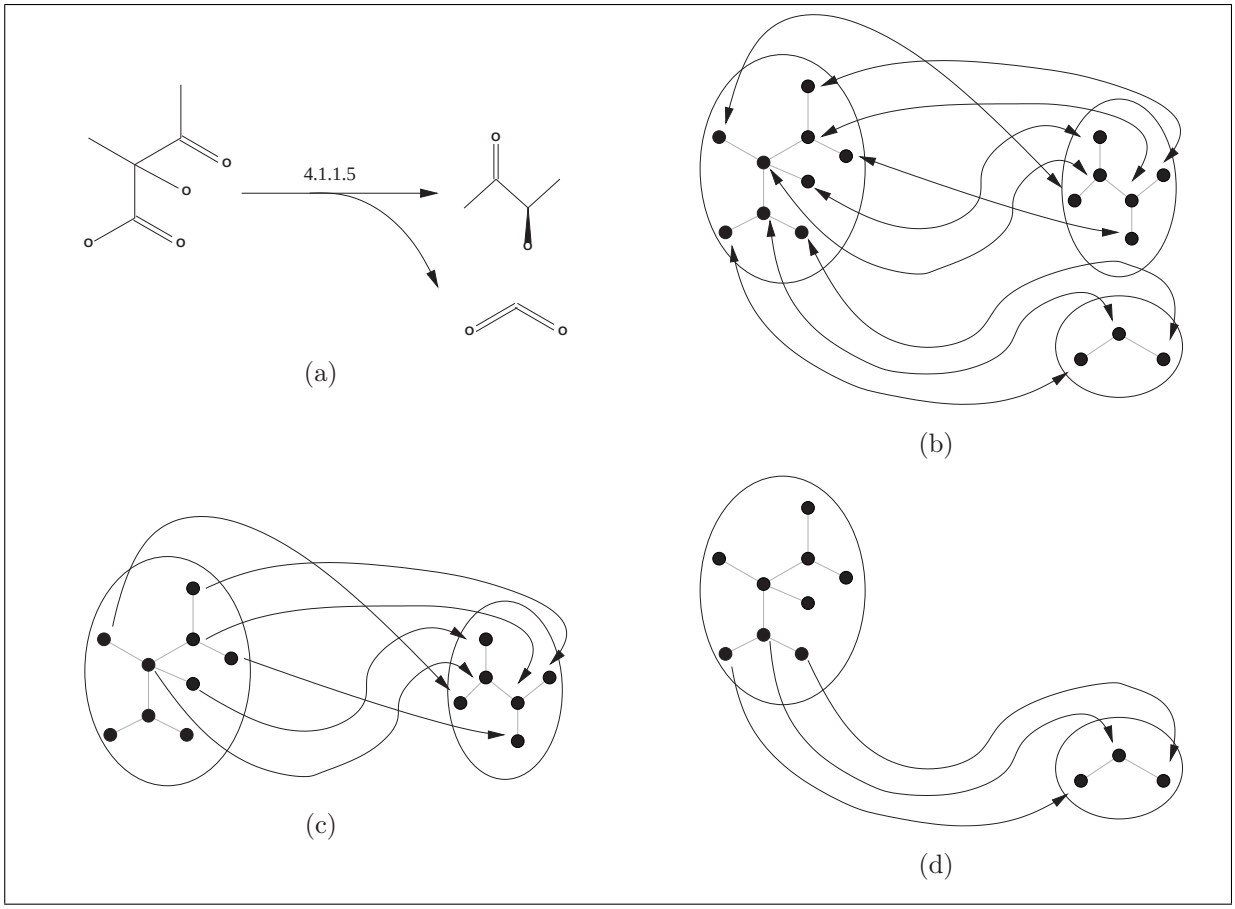


Fig. 8.2: Extraction de fonctions injectives partielles entre les atomes de chaque couple de composés à partir d'une correspondance atomique pour une réaction - (a) montre une réaction pour laquelle la correspondance atomique (b) est connue. A partir de cette correspondance atomique deux injections partielles (c) et (d) sont extraites pour chaque couple de composés $(substrat, produit)$

Sur la base de ces injections, il est ensuite possible de calculer le transfert d'atomes pour n'importe quelle succession de réactions entre deux composés.

L'approche peut donc être décomposée en deux étapes :

1. définir une correspondance entre atomes pour chaque couple de composés $(substrat, produit)$ d'une réaction (à effectuer une fois pour toutes pour chaque réaction de la base)
2. calculer, sur la base de ces correspondances, toutes les successions de réactions assurant le transfert d'un nombre minimum d'atomes entre deux composés (à effectuer pour chaque nouvelle requête).

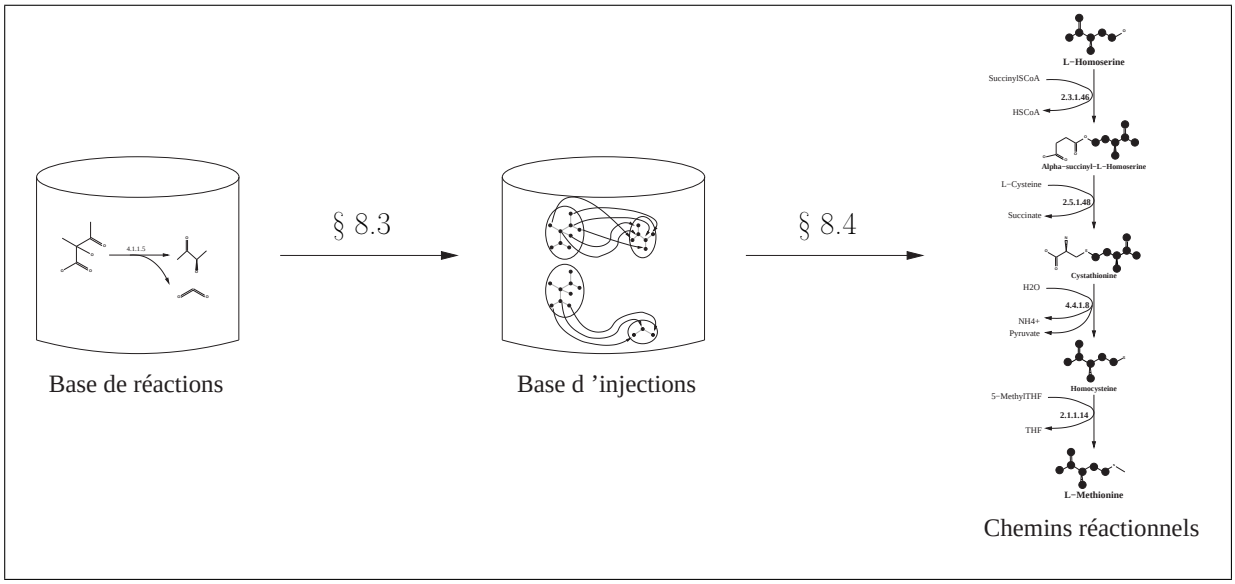


Fig. 8.3: La seconde approche explorée pour la reconstruction *ab initio* peut être décomposée en deux étapes successives - ces deux étapes sont respectivement traitées aux § 8.3 et § 8.4

8.3 Recherche des correspondances atomiques induites par les réactions

Le problème de la recherche des correspondances entre atomes induites par une réaction est un problème qu'il est possible de poser comme la recherche du Sous-graphe Commun Maximal de deux graphes, chacun de ces deux graphes correspondant à un côté de la réaction. Ce problème a déjà été décrit au § 5.4.2 (problème 6 - page 80).

Au moins deux algorithmes ont été spécialement conçus pour résoudre ce problème dans le cas des réactions enzymatiques (décrits au § 5.4 - page 77). Le premier algorithme [Arita, 2000a] est un algorithme heuristique glouton qui se base sur la recherche des sous-structures communes de taille maximale entre couples de composés. Le second algorithme [Akutsu, 2003] résout le problème de manière exacte pour des cas particuliers de réactions (voir § 5.4.2.1). Dans les deux cas, les graphes utilisés pour représenter les structures des composés ne font pas la distinction entre les types de liaisons chimiques

entre les atomes (seul les nœuds sont étiquetés avec le type d'atome correspondant, les arêtes ne sont pas étiquetées). Cela vient du fait que des liaisons sont souvent modifiées au cours d'une réaction et que ce type d'événement n'est pas aussi important que la suppression/création d'une liaison. Aussi, pour la suite de cette section nous considérons également les arêtes des graphes moléculaires comme non étiquetées.

Dans les paragraphes suivants, nous rappelons brièvement le lien établi au § 5.4.2 entre le problème de la recherche des correspondances entre atomes induites par une réaction et le problème du Sous-graphe Commun Maximal. Ensuite, une extension de l'algorithme glouton est présentée. Enfin, nous présentons les résultats de l'application de l'algorithme glouton et du nouvel algorithme sur les réactions de la banque LIGAND/KEGG.

8.3.1 Problème du Sous-graphe Commun Maximal de deux graphes

8.3.1.1 Lien avec le problème du calcul de la correspondance atomique induite par une réaction

Comme nous l'avons vu au § 5.4.2, la recherche de la correspondance atomique induite par une réaction consiste à rechercher un couplage entre les deux ensembles de nœuds des deux graphes représentant les deux côtés de la réaction. Le couplage doit bien entendu tenir compte de l'étiquetage des nœuds.

Exemple : la figure 8.4(a) montre une réaction avec, de chaque côté de la réaction, le graphe moléculaire correspondant aux composés impliqués dans cette réaction. Les nœuds des deux graphes ont été numérotés pour plus de clarté. La figure 8.4(b) montre le graphe bipartite dans lequel il faut choisir le couplage. A chaque couplage de taille maximale est associée une correspondance entre les atomes des substrats et des produits.

Il est aisé de dénombrer le nombre de couplages de taille maximale possible. Si on note N_x le nombre de nœuds associés à l'étiquette x, alors on a

$$\prod_{x \in \{C,O,N,P,Fe,Mg,\ldots\}} N_x!$$

couplages différents de taille maximale. Pour l'exemple de la figure 8.4, cela fait approximativement 10^{17} couplages.

Parmi tous ces couplages, on va chercher ceux qui conservent le plus grand nombre de liaisons chimiques entre les atomes (c'est-à-dire ceux qui cassent le moins de liaisons). Ainsi, les couplages C (de taille maximale) recherchés doivent donc maximiser la fonction suivante :

$$\sum_{(v_1,v_2),(v'_1,v'_2) \in C^2} (\text{coût}((v_1, v_2), (v'_1, v'_2)))$$

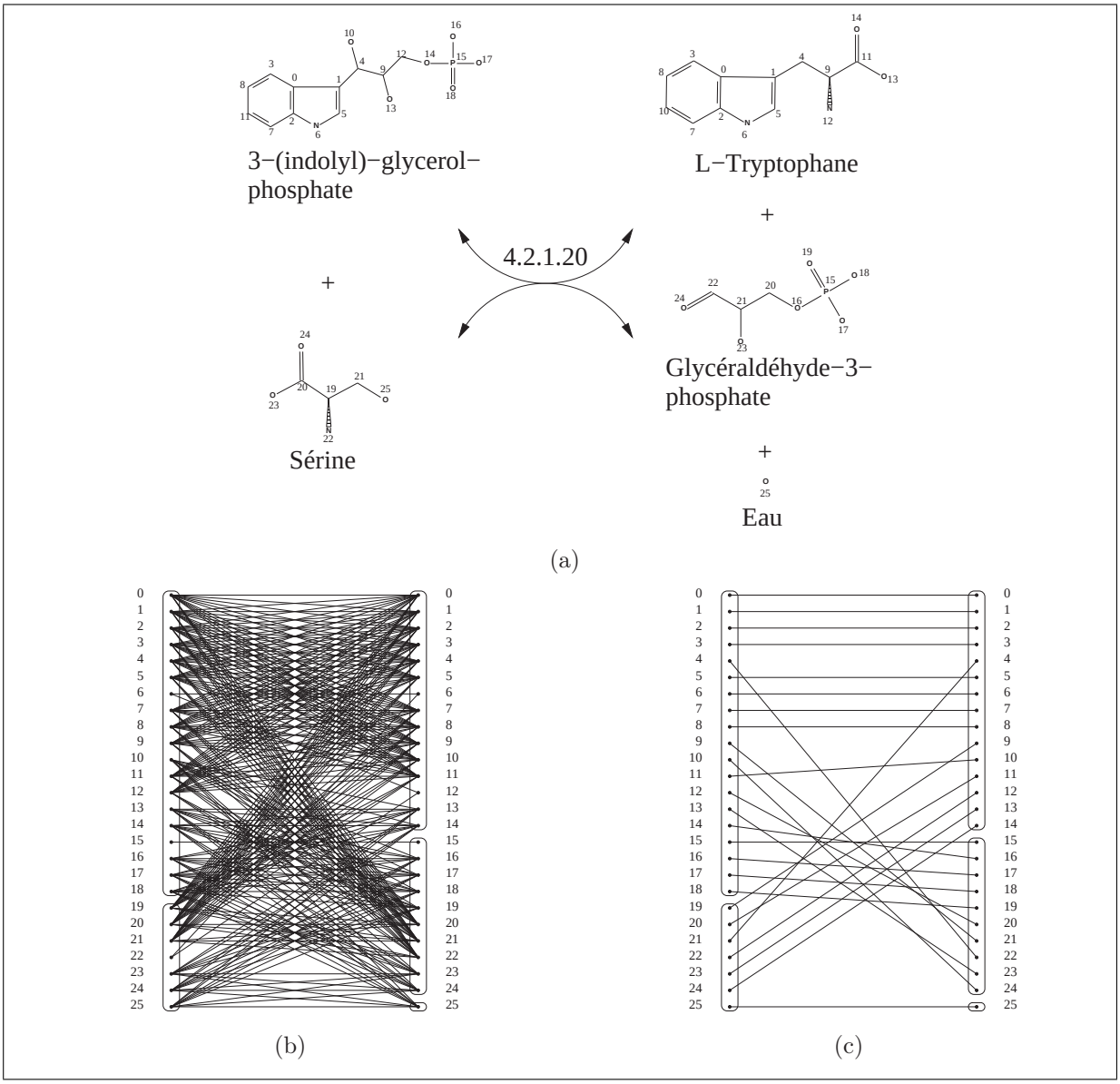


Fig. 8.4: Une réaction (a), le graphe bipartite des correspondances atomiques associé (b) et un couplage maximal possible (c)

où coût$((v_1, v_2), (v'_1, v'_2))$ vaut :

$$\begin{cases} 1 & \text{si } v_1 \text{ et } v'_1 \text{ sont des nœuds reliés dans le graphe des substrats et} \\ & \text{si } v_2 \text{ et } v'_2 \text{ sont des nœuds reliés dans le graphe des produits} \\ 0 & \text{sinon} \end{cases}$$

Ce problème peut donc être posé dans les termes du Sous-graphe Commun Maximal, présenté au § 5.4.2, dans lequel l'isomorphisme est donné par le couplage.

8.3.1.2 Résolution du problème Sous-graphe Commun Maximal par une recherche de clique de poids maximal dans un graphe pondéré

Il est possible de résoudre de manière exacte le problème du Sous-graphe Commun Maximal en résolvant un autre problème nommé Clique de Poids Maximal. Ce problème est défini ci-après.

Définition **19** *Clique*
Une clique est un graphe $K = (V, E)$ tel que :

$$\forall\, (u, v) \in V, (u, v) \in E$$

"Graphe complet" est un synonyme de clique, on note K_x le graphe complet à x nœuds ($|V| = x$).

Problème **11** Clique de Poids Maximal
Données : *un graphe $\mathcal{G} = (V, E)$ et une fonction de pondération des arêtes $p_E : E \to \mathbb{N}$*
Réponse : *un sous-graphe complet $K = (V', E')$ de $\mathcal{G}$*
Mesure : $\sum_{e \in E'} p_E(e)$
Optimisation : max

Il nous faut maintenant décrire le graphe dans lequel nous recherchons la clique de poids maximal qui correspond à la solution du problème Sous-graphe Commun Maximal.

Ce graphe, $\mathcal{G}$, est défini de la façon suivante, à partir des deux graphes $\mathcal{G}_A = (V_A, E_A)$ et $\mathcal{G}_B = (V_B, E_B)$ qui sont les données du problème Sous-graphe Commun Maximal :

- $V = \{(v_A, v_B) \in V_A \times V_B / \text{étiquette}\,(v_A) = \text{étiquette}\,(v_B)\}$
- $E = \{(v = (v_A, v_B), u = (u_A, u_B)) \in V \times V / v_A \neq u_A \text{ et } v_B \neq u_B\}$
- $\forall e = (v = (v_A, v_B), u = (u_A, u_B)) \in E,$

$$p_E(e) = \begin{cases} 1 \text{ si} & (v_A, u_A) \in E_A \text{ et } (v_B, u_B) \in E_B, \text{ ou} \\ & (v_A, u_A) \notin E_A \text{ et } (v_B, u_B) \notin E_B \\ 0 \text{ sinon} & \end{cases}$$

En d'autres termes, les nœuds de $\mathcal{G}$ sont des couples d'atomes (un atome pour chaque côté de la réaction) de même étiquette. Deux nœuds sont reliés s'ils ne partagent pas d'atome commun. Les arêtes sont pondérées par 0 ou 1. La pondération vaut 1 lorsque dans chaque graphe originel, les deux couples d'atomes sont simultanément soit connectés, soit non connectés.

Ceci est illustré sur la figure 8.5.

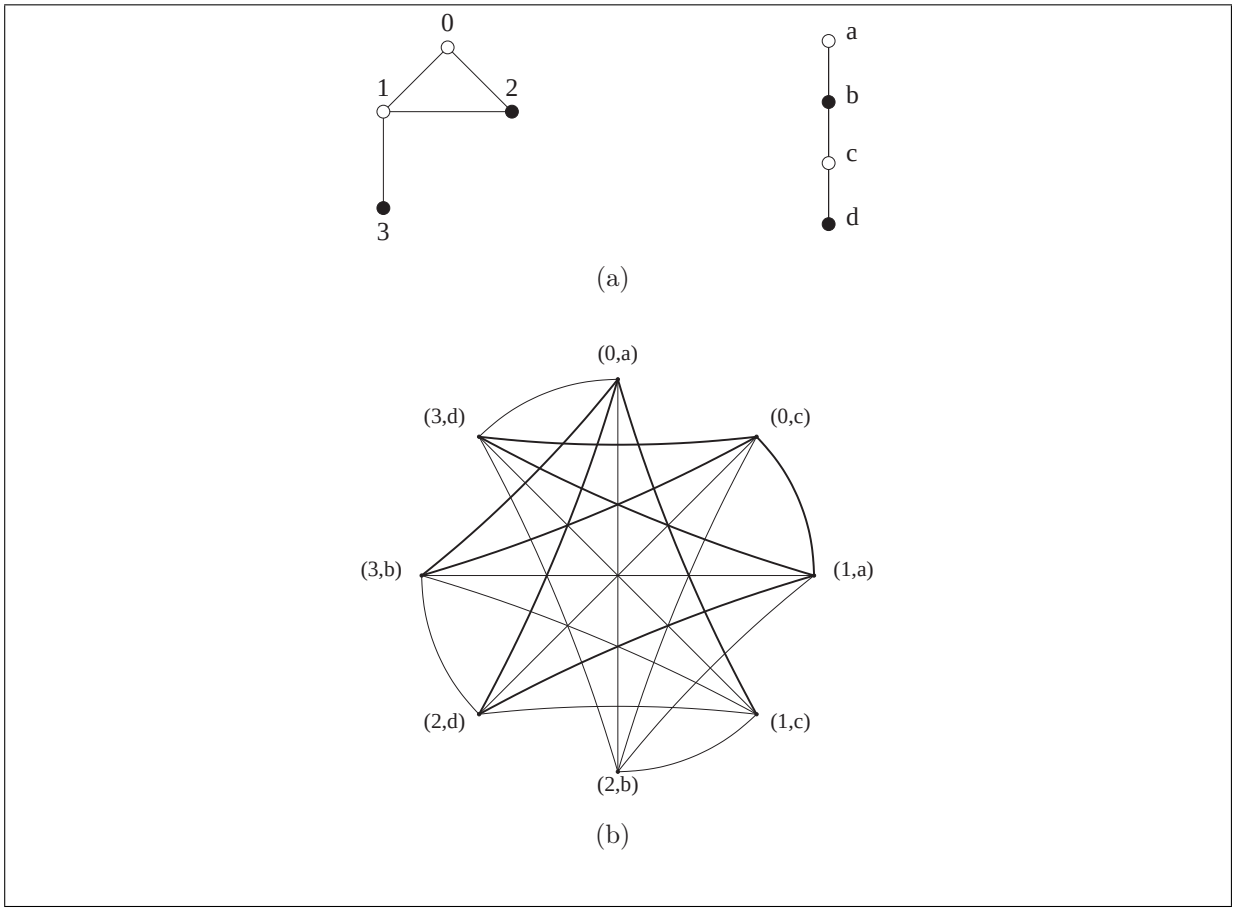


Fig. 8.5: Instance du problème Sous-graphe Commun Maximal - L'instance du problème Sous-graphe Commun Maximal est donnée en (a), les deux graphes ont leurs nœuds étiquetés par l'étiquette 'noir' ou 'blanc', (b) montre le graphe $\mathcal{G}$ construit à partir de cette instance

Dans $\mathcal{G}$, chaque clique de taille maximale correspond un couplage entre les nœuds de $\mathcal{G}_A$ et $\mathcal{G}_B$. La clique de taille maximale qui a le plus fort poids correspond au couplage de coût maximal[5] (la solution optimale du problème Sous-graphe Commun Maximal pour les graphes $\mathcal{G}_A$ et $\mathcal{G}_B$).

[5]On notera qu'il n'y a pas de distinction entre l'absence de liaison entre deux atomes d'un même composé et l'absence de liaison entre deux atomes de deux composés différents. De plus, il est possible que la clique de poids maximal du graphe $\mathcal{G}$ ne soit pas également de taille maximale et ne corresponde

Exemple : la figure 8.6(c) montre le graphe construit à partir des deux graphes de la figure 8.6(a). Les arêtes ayant un poids de 0 sont en gras. Dans ce graphe, il y a 4 cliques de taille maximale, et chacune d'elles correspond à un couplage possible. La clique $\{(0,a),(1,c),(2,b),(3,d)\}$ (figure 8.6(d)) est la clique de poids maximal et correspond à la solution optimale du problème original représenté sur la figure 8.6(b).

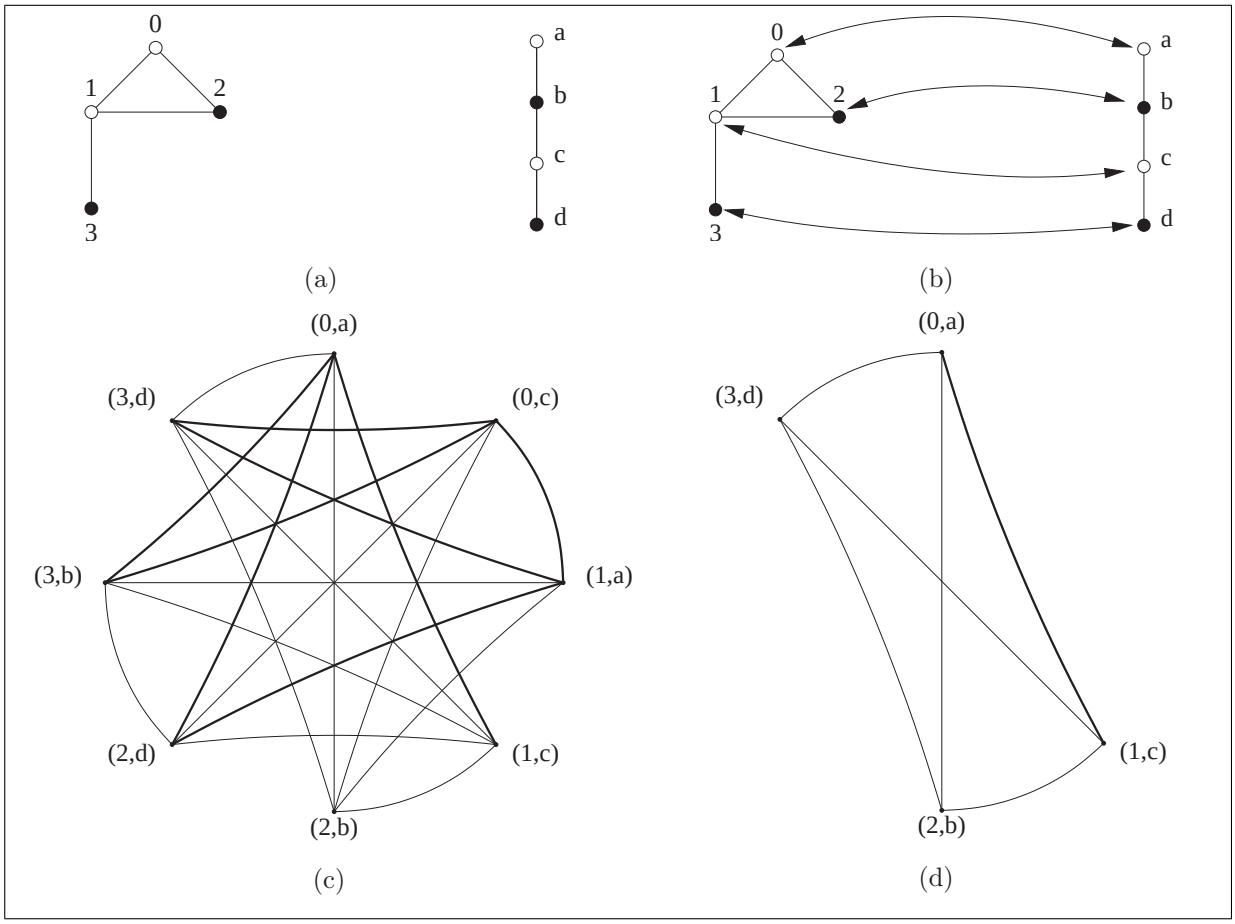


Fig. 8.6: Instance et solution pour le problème Sous-graphe Commun Maximal - L'instance du problème Sous-graphe Commun Maximal est donnée en (a), les deux graphes ont leurs nœuds étiquetés par l'étiquette 'noir' ou 'blanc', (b) montre la solution optimale de cette instance. (c) montre le graphe $\mathcal{G}$ construit à partir de cette instance, les traits fins correspondent à une pondération de 1 et les traits gras correspondent à une pondération de 0, (d) montre la clique (de taille maximale et) de poids maximal correspondant à la solution optimale montrée en (b)

La taille du graphe $\mathcal{G}$ est de l'ordre de $\mathcal{O}(n^2)$ où n est la taille des deux graphes donnés en entrée du problème Sous-graphe Commun Maximal. Comme le problème Clique

donc pas à un couplage complet entre les nœuds des graphes $\mathcal{G}_A$ et $\mathcal{G}_B$. Dans ce cas, il existe toujours une clique de taille maximale de même poids incluant la clique de plus fort poids.

de Poids Maximal est un problème difficile ($\mathcal{NP}$-difficile [Garey and Johnson, 1979]), et compte tenu de la croissance de la taille du graphe $\mathcal{G}$, la résolution exacte du problème de cette façon est possible seulement avec des graphes de petite taille (moins de quelques dizaines d'atomes). Dans le cas général, il faut donc recourir à des méthodes de résolutions heuristiques.

8.3.2 Un nouvel algorithme heuristique : extension de l'algorithme glouton basé sur la recherche des sous-structures communes entre composés

L'idée de cette heuristique est de travailler non sur la totalité du graphe $\mathcal{G}$, mais de le partitionner en graphes plus petits. Il s'agit alors d'optimiser localement la fonction objective (sur les petits graphes) et de composer les résultats. C'est le cas de l'algorithme glouton présenté au § 5.4.2.2 - page 83 dont le principe est de considérer dans la solution finale l'appariement correspondant à la sous-structure de taille maximale entre tous les couples de composés pris individuellement.

Dans bien des cas, cet algorithme donne de bons résultats (proche ou égal à l'optimal), cependant pour l'exemple de la figure 8.7, l'algorithme glouton ne donnera pas la solution optimale. En effet, dans ce cas, la plus grande sous-structure commune aux couples de composés pris individuellement (indiquée en traits gras) ne correspond pas à la solution optimale indiquée par les boites.

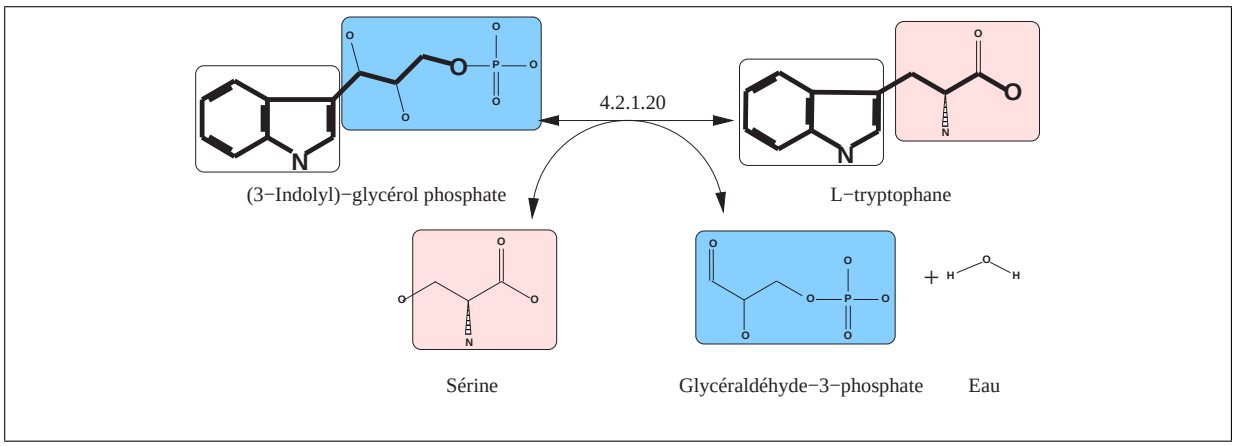


Fig. 8.7: Un exemple de réaction où l'affectation de la plus grande sous-structure commune entre couples de composés $(substrat, produit)$ est en contradiction avec la solution optimale du problème Sous-graphe Commun Maximal

L'espace de recherche lié cette résolution est représenté sur la figure 8.8.

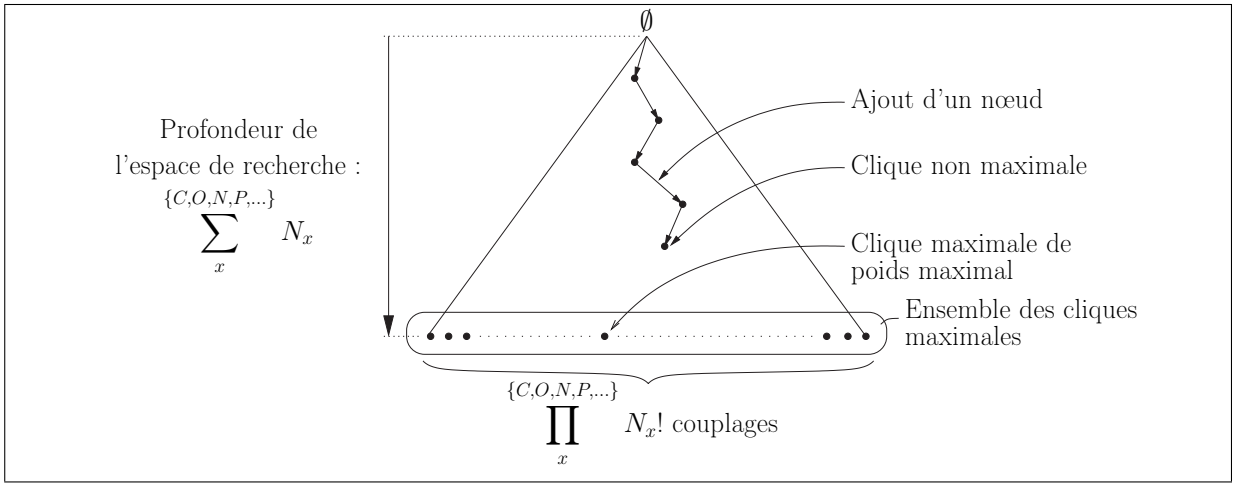


Fig. 8.8: Espace de recherche du problème Sous-graphe Commun Maximal

Dans l'algorithme glouton, la sélection d'une sous-structure correspond à un saut dans l'espace de recherche associé à une optimisation locale du critère objectif. Le critère à optimiser étant la taille des sous-structures, les bonds successifs dans l'espace de recherche sont de plus en plus petits. Comme l'algorithme est glouton, un seul chemin dans l'espace de recherche est exploré (la figure 8.9 représente l'exploration effectuée par l'algorithme glouton).

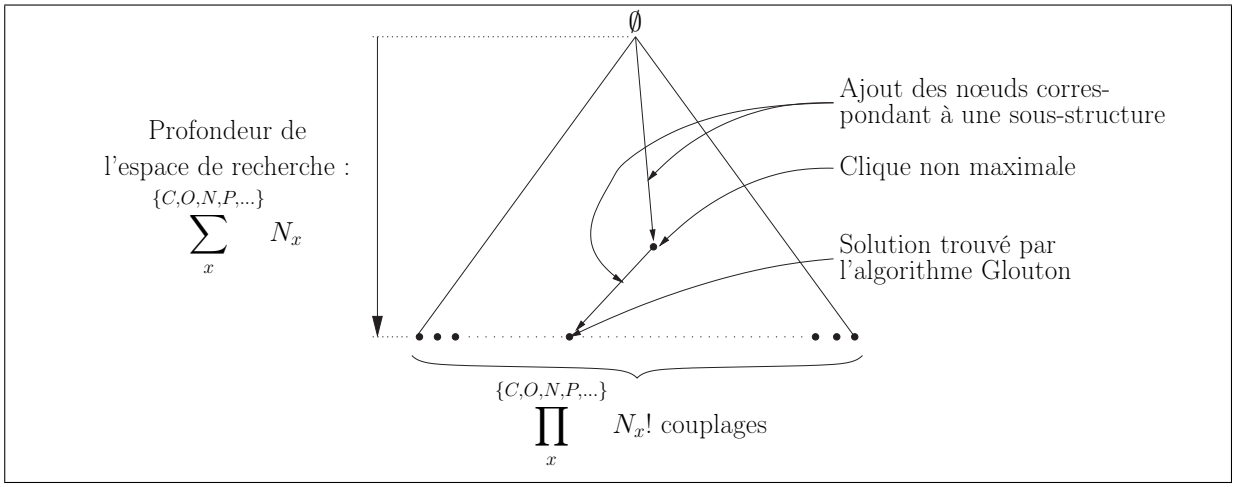


Fig. 8.9: Illustration de l'exécution de l'algorithme glouton par rapport à l'espace de recherche du problème Sous-graphe Commun Maximal

Nous proposons un nouvel algorithme basé sur l'algorithme glouton qui dans le cas précis de l'exemple de la figure 8.7, donne un meilleur résultat.

8.3.2.1 Extension de l'algorithme glouton

L'idée de notre extension est, comme pour l'algorithme glouton, de se restreindre aux couples de composés mais de ne pas nécessairement sélectionner la plus grande sous-structure maximale à chaque étape de l'exploration (cette approche est similaire à une descente en "beam search").

Ceci est illustré sur la figure 8.10.

8.3.2.2 Mise en œuvre

Cet algorithme a été mis en œuvre en langage *C* dans une version utilisant la stratégie d'énumération A* :

La fonction d'évaluation Elle se contente de compter le nombre d'arêtes non-conservées lors de l'assignation des sous-structures.

Fonction d'estimation Une borne inférieure au nombre d'arêtes qui vont être supprimées à partir d'une assignation incomplète peut être obtenue en comptant pour les deux ensembles de nœuds en attente d'assignation, le nombre de composantes connexes dans les sous-graphes restants. La fonction d'estimation vaut alors $|NbComposantesConnexes(\mathcal{G}_1) - NbComposantesConnexes(\mathcal{G}_2)|$.

8.3.3 Application aux données de la banque LIGAND/KEGG

L'intégralité des réactions de la base LIGAND/KEGG a d'abord été traitée par l'algorithme glouton. Dans le cas où le nombre de liaisons chimiques non conservées était supérieur au nombre de composés impliqués dans la réaction, le nouvel algorithme a été appliqué pour améliorer ce score.

4410 réactions ont été traitées avec l'algorithme glouton. Pour 25 de ces réactions, les calculs n'ont pas aboutis car le temps écoulé était excessif (au delà de 1 heure de calcul par réaction). Pour la grande majorité des réactions le temps de calculs est néanmoins faible, moins de 10 secondes pour plus de 4000 réactions.

248 réactions ont été sélectionnées pour être soumises au nouvel algorithme. Pour 55 réactions, le score a été amélioré.

Le figure 8.11 représente l'histogramme des scores obtenus pour les 4385 réactions effectivement traitées.

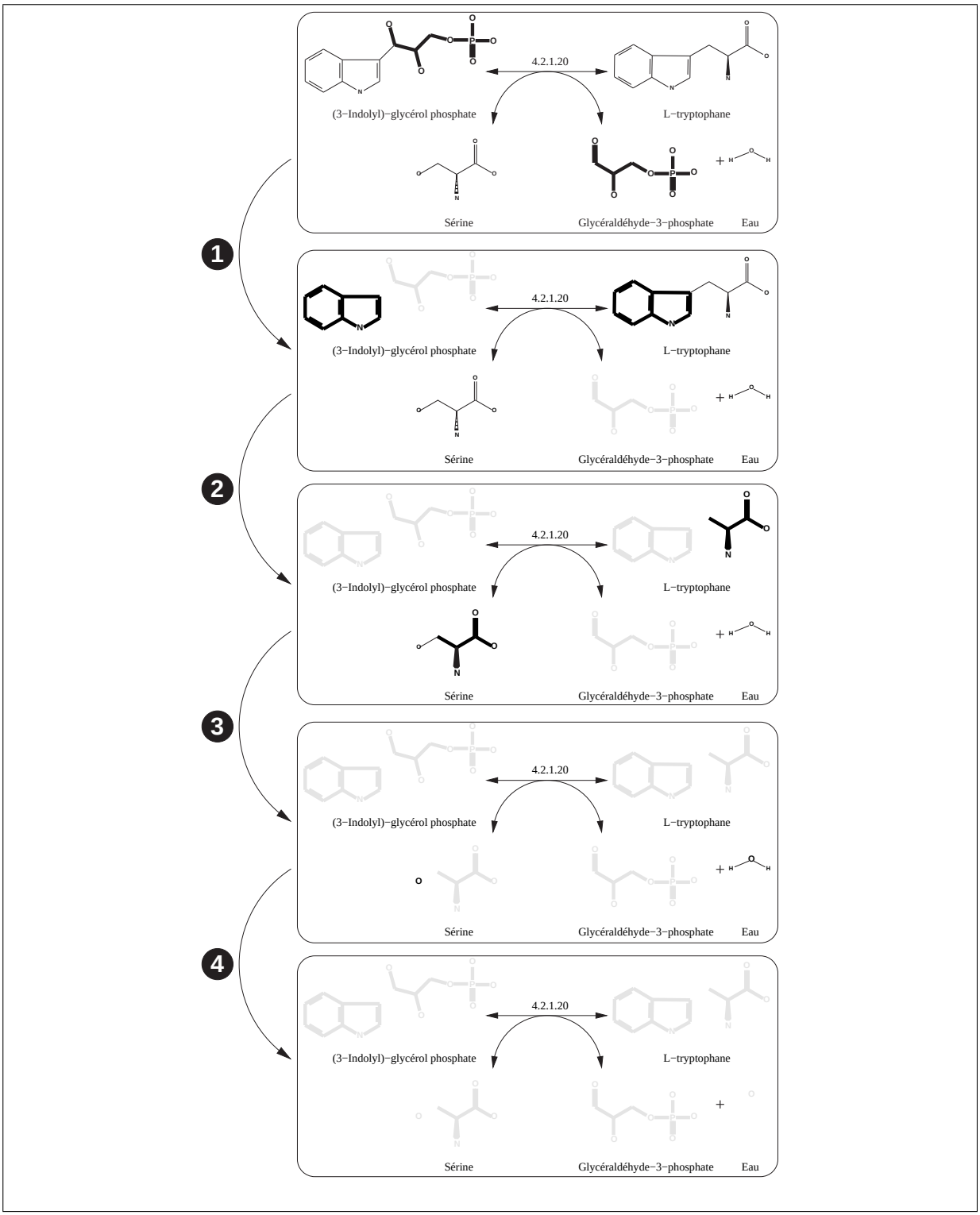


Fig. 8.10: Appariement optimal (au sens du problème Sous-graphe Commun Maximal) des atomes des composés de la réaction d'EC 4.2.1.20. Chacune des étapes consiste en la mise en correspondance des nœuds composant une des sous-structures connexes de taille maximale (représentée en gras à chaque étape)

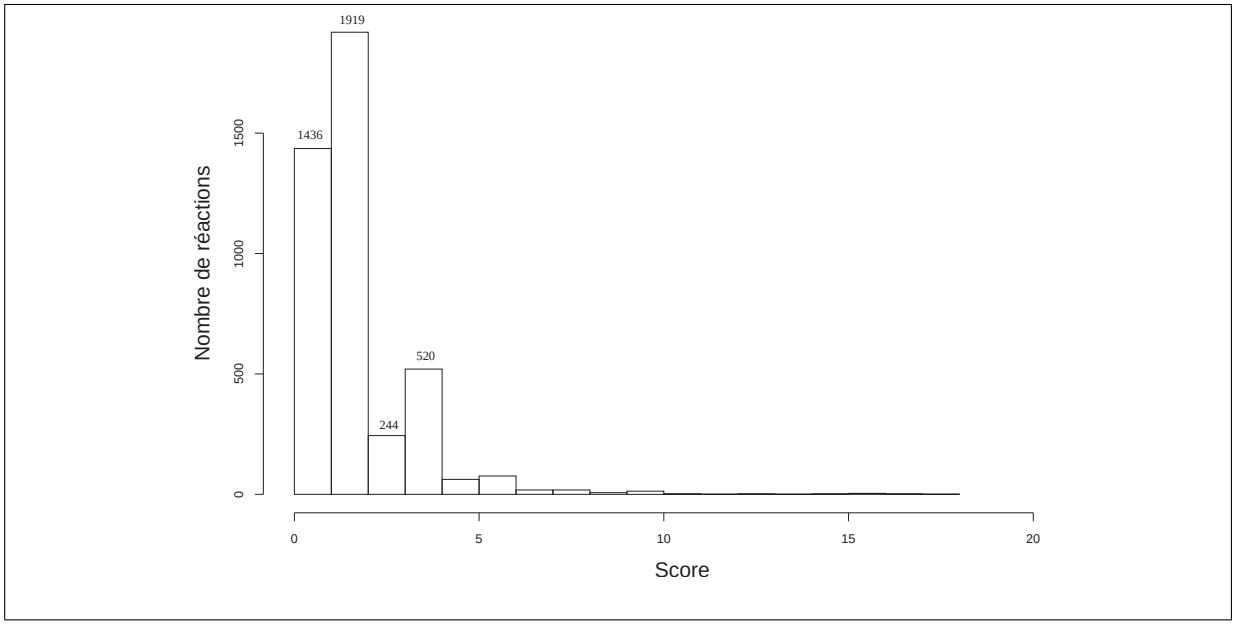


Fig. 8.11: Nombre de réactions en fonction du nombre de liaisons chimiques supprimées pour établir la correspondance des atomes entre composés

A partir de ces correspondances 12695 couplages entre composés ont été extraits. Cela définit 8089 couplages différents car un même couplage peut être issu de plusieurs réactions. La figure 8.12 donne la répartition des tailles des différents couplages.

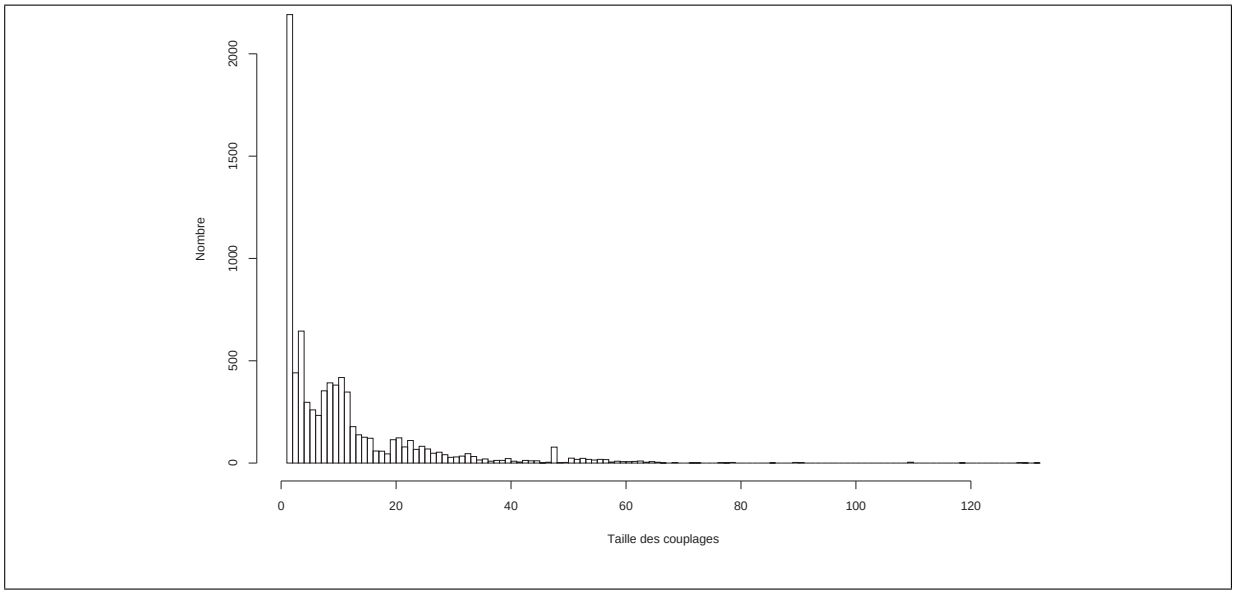


Fig. 8.12: Histogramme des tailles des couplages atomiques entre les composés

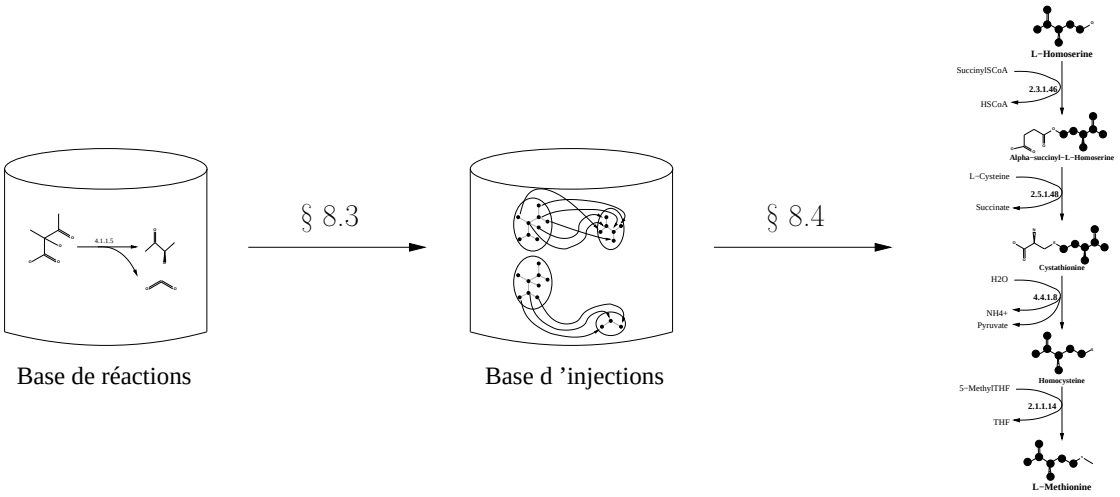
La plupart de ces couplages (7040 sur les 8089) sont issus d'une seule réaction, d'autres en revanche sont induits par plusieurs réactions. Les couplages issus du plus grand nombre de réactions concernent les couples de composés suivants :

– $NADP^+$ et NADPH : 572 fois
– NAD^+ et NADH : 512 fois
– ATP et ADP : 241 fois

Cela n'a rien d'étonnant, ces couples de composés servant d'apport énergétique aux réactions.

8.4 Recherche des chemins réactionnels entre deux composés

La recherche des chemins réactionnels entre deux composés est le problème qui a motivé la recherche des correspondances atomiques entre les composés. En effet, comme le rappelle le schéma suivant, la recherche des correspondances atomiques entre les composés est une étape nécessaire pour résoudre le problème de la recherche des chemins réactionnels tel que nous l'avons présenté. Ce travail a fait l'objet d'une publication donnée à l'annexe A.



Une fois que les correspondances entre atomes pour toutes les réactions sont disponibles et que les injections partielles entre les atomes des couples de composés $(substrat, produit)$ ont été extraites, le problème de la reconstruction *ab initio* de voies métaboliques peut être formulé comme un problème d'optimisation basé sur la composition de fonctions injectives partielles.

Dans ce chapitre, nous formulons explicitement ce problème, puis nous analyserons sa complexité. Enfin, un algorithme le résolvant exactement est présenté.

8.4.1 Formulation

Comme énoncé précédemment, les correspondances atomiques entre les couples de composés $(substrat, produit)$ des réactions peuvent être manipulées comme des fonctions injectives partielles entre les deux ensembles d'atomes des composés. Il est possible de composer deux injections partielles, le résultat d'une telle composition est lui-même une injection partielle (voir figure 8.13).

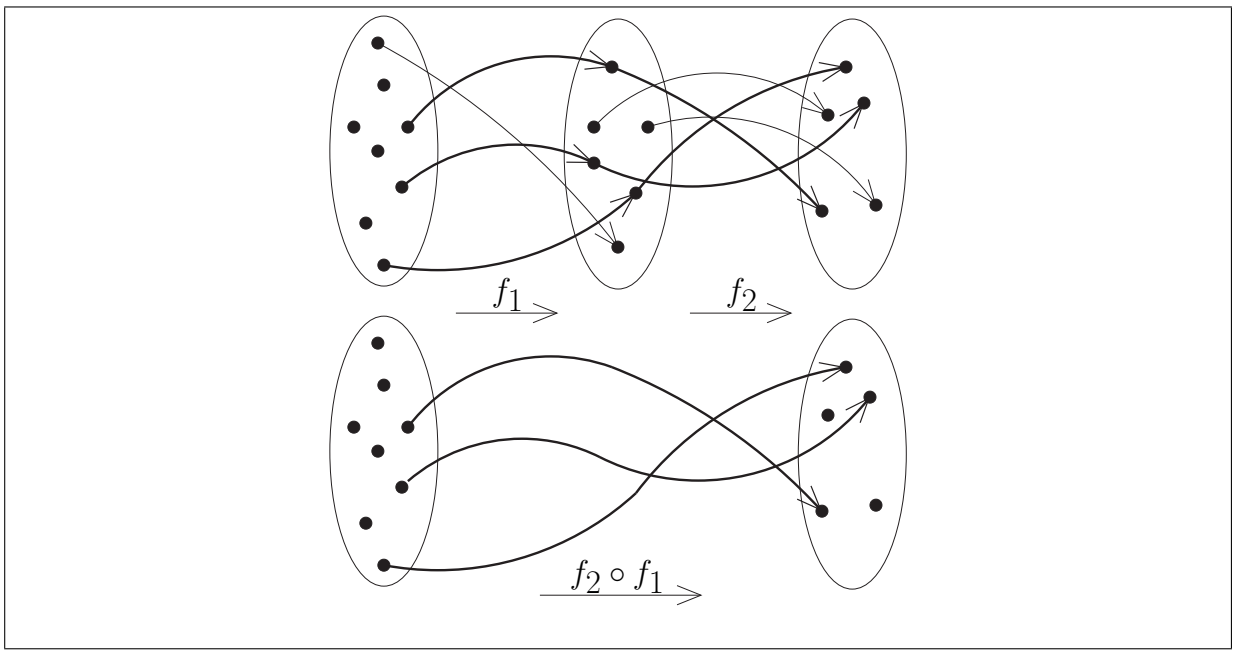


Fig. 8.13: La composition de deux injections partielles est une injection partielle - L'injection partielle résultant de la composition ne met en correspondance que 3 éléments bien que les deux injections f_1 et f_2 mettaient respectivement 4 et 5 éléments en correspondance

De cette façon, n'importe quel chemin réactionnel peut être représenté par une composition de fonctions injectives partielles.

Exemple : la figure 8.14 donne un exemple de chemin réactionnel, entre la L-homosérine et la L-méthionine et de la succession de réactions correspondante.

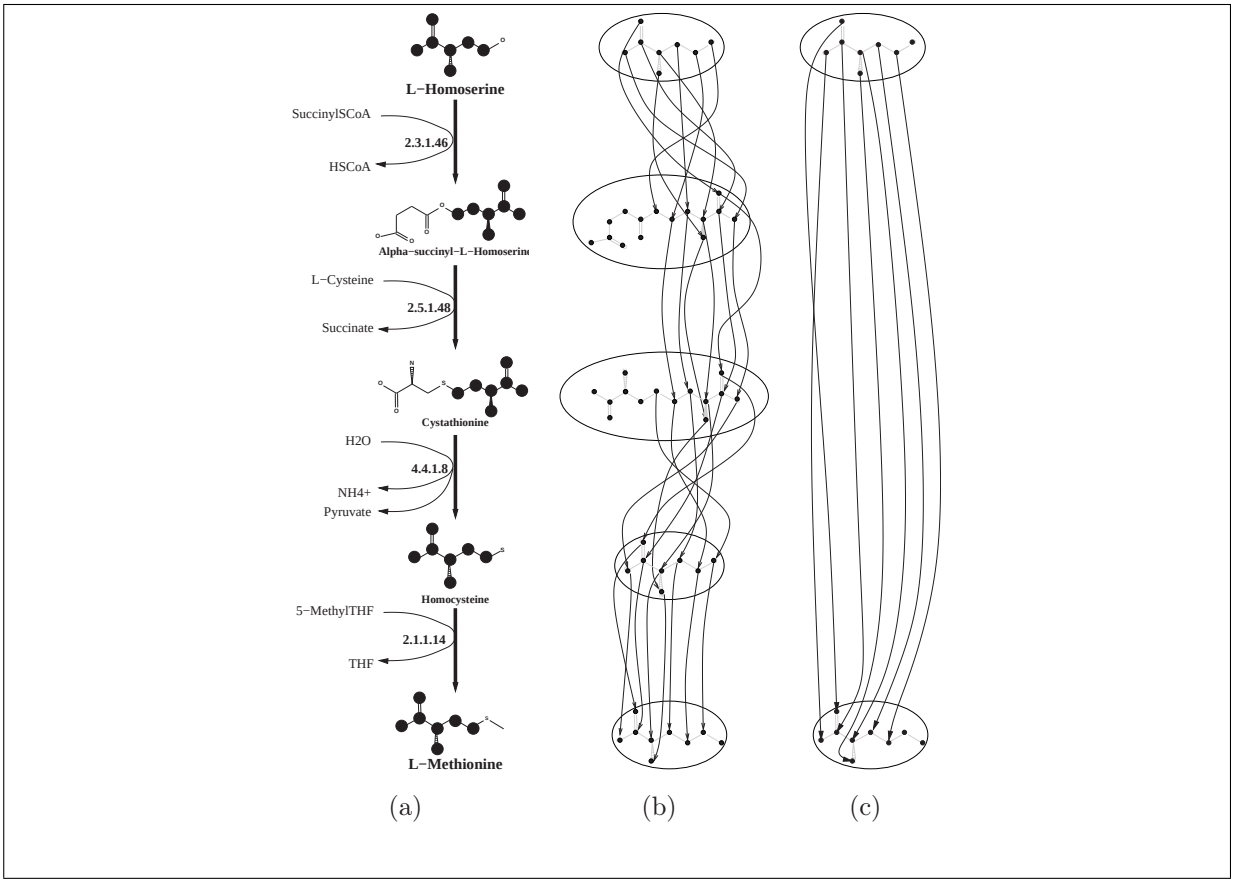


Fig. 8.14: Exemple de chemin réactionnel et de la composition de fonctions injectives partielles correspondante - le chemin réactionnel de la L-Homosérine à la L-Méthionine (a) peut être représenté sous la forme d'une suite de fonctions injectives partielles (b) dont la composition est représentée en (c)

Le nombre total d'atomes transférés entre le substrat initial et le produit final est la taille de l'image de la composition résultat. Si l'on souhaite rechercher les chemins réactionnels qui maximisent la conservation des atomes entre le composé initial et le composé final, le problème de la recherche d'un chemin entre deux composés peut être formulé de la façon suivante :

Problème **12** Composition Maximale d'Injections Partielles (MPIC)
Données : *un ensemble de n ensembles distincts $\mathcal{X} = \{X_1, \ldots, X_n\}$, un ensemble de m injections partielles (définies sur les ensembles de $\mathcal{X}$) $\mathcal{I} = \{I_1, \ldots, I_m\}$, une paire d'ensembles $(X_i, X_j) \in \mathcal{X}^2$*
Réponse : *une composition d'injections partielles $I_{comp} \in \mathcal{I}^*$ de X_i à X_j*
Mesure : $Taille\,(I_{comp})$
Optimisation : max
Note : étant donnée une injection partielle $I : X \to Y$, on définit sa taille par $Taille(I) =$

$|I(X)|$, *c'est-à-dire le nombre de* $x_i \in X$ *qui ont une image dans* Y *par* I.

8.4.2 Complexité du problème MPIC

Afin de caractériser le problème MPIC, sa complexité est étudiée (le lecteur peu habitué à la notion de complexité peut se reporter à l'annexe C).

Pour établir la complexité du problème MPIC, il faut étudier la complexité du problème de décision associé.

Problème **13** Composition d'Injections Partielles (PIC)
Données : *un ensemble de* n *ensembles distincts* $\mathcal{X} = \{X_1, \ldots, X_n\}$, *un ensemble de* m *injections partielles (définies sur les ensembles de* $\mathcal{X}$*)* $\mathcal{I} = \{I_1, \ldots, I_m\}$, *une paire d'ensembles* $(X_i, X_j) \in \mathcal{X}^2$
Question : *existe-t-il une composition d'injections partielles* $I_{comp} \in \mathcal{I}^*$ *de* X_i *à* X_j *telle que* $Taille(I_{comp}) = \min(|X_i|, |X_j|)$ *?*

Proposition **1** *Le problème PIC est* $PSPACE$*-Complet.*

Pour prouver que le problème PIC est $PSPACE$-Complet, les instances d'un problème voisin du problème Intersection d'Automates d'Etats Finis sont réduites aux instances du problème PIC. Le problème Intersection d'Automates d'Etats Finis a été prouvé comme faisant partie des problèmes $PSPACE$-Complets [Kozen, 1977]. Pour la réduction, ce sont les instances du problème Intersection d'Automates Injectifs, un problème moins général que Intersection d'Automates d'Etats Finis, qui sont réduites aux instances de PIC. Le problème Intersection d'Automates Injectifs a été prouvé comme faisant également partie des problèmes $PSPACE$-Complets [Birget *et al.*, 2000].

Preuve **1** *Comme il est possible de "deviner" une séquence* $s \in \mathcal{I}^*$, *d'appliquer les injections de cette séquence sur les ensembles résultats successifs en commençant avec* X_i, *et de répondre non si il est impossible de composer deux injections successives (parce que les domaines ne coïncident pas) ou si une taille trop faible est atteinte, ou oui sinon, alors le problème PIC appartient la classe de problèmes* $NPSPACE$. *Comme tous les problèmes de NPSPACE font également partie de* $PSPACE$ *[Savitch, 1970], PIC appartient à* $PSPACE$.

Définition ***20*** *Automate injectif*
Soit $\mathcal{A} = (Q, \Sigma, \delta, i, F)$ *un automate d'états finis déterministe défini par son ensemble d'états* Q, *son ensemble fini de symboles* Σ, *sa fonction de transition* $\delta : Q \times \Sigma \to Q$, *son état initial* i *et son ensemble d'états finaux* $F \subseteq Q$.

$\mathcal{A}$ est injectif si et seulement si chaque symbole $\sigma \in \Sigma$ induit une fonction injective partielle de Q sur Q et $|F| = 1$.

Exemple : L'automate de la figure 8.15(a) est injectif alors que l'automate 8.15(b) n'est pas injectif.

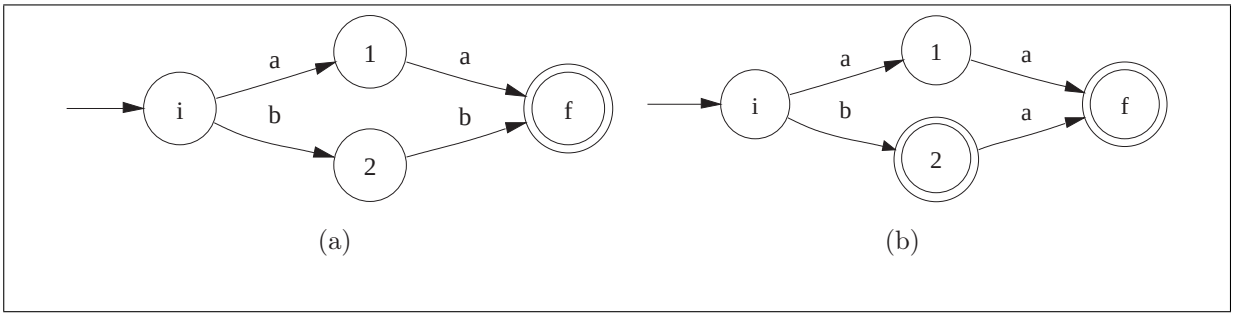


Fig. 8.15: Exemple d'automates injectifs et non injectifs - L'automate (a) est injectif tandis que l'automate (b) n'est pas injectif pour deux raisons : il a deux états finaux et deux arcs étiquetés avec le même symbole arrivent à l'état f

Le problème est le suivant :

Problème ***14*** Intersection d'Automates Injectifs
Instance *: un ensemble $\{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ d'automates injectifs ayant le même alphabet*
Question *: existe-t-il une chaîne $x \in \Sigma^*$ acceptée par chacun des automates $\mathcal{A}_i$, $1 \leq i \leq n$?*

Théorème ***1*** *Le problème* Intersection d'Automates Injectifs *est PSPACE-Complet [Birget* et al.*, 2000]*

La réduction, illustrée sur la figure 8.16, est la suivante :

soit $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ un ensemble de n automates injectifs partageant le même alphabet $\Sigma = \left\{\sigma_1, \ldots, \sigma_{|\Sigma|}\right\}$, avec $\mathcal{A}_i = \left(Q_{\mathcal{A}_i}, \Sigma, \delta_{\mathcal{A}_i}, q_{\mathcal{A}_i}^{init}, \left\{q_{\mathcal{A}_i}^{final}\right\}\right)$, $Q_{\mathcal{A}_i} = \left\{q_{\mathcal{A}_i}^1, \ldots, q_{\mathcal{A}_i}^{|Q_{\mathcal{A}_i}|}\right\}$, $q_{\mathcal{A}_i}^{init} \in Q_{\mathcal{A}_i}$ et $q_{\mathcal{A}_i}^{final} \in Q_{\mathcal{A}_i}$

on définit trois ensembles X_1, X_2 et X_3 :

- $X_1 = \{x_1^1, \ldots, x_1^n\}$
- $X_2 = \bigcup_{i=1}^{n} Q_{\mathcal{A}_i}$
- $X_3 = \{x_3^1, \ldots, x_3^n\}$

ainsi que $2 + |\Sigma|$ injections partielles :

- *I_{init} de X_1 à X_2 : $I_{init}(x_1^i) = q_{\mathcal{A}_i}^{init}$, $1 \leq i \leq n$*
- *I_{final} de X_2 à X_3 : $I_{final}\left(q_{\mathcal{A}_i}^{final}\right) = x_3^i$, $1 \leq i \leq n$*

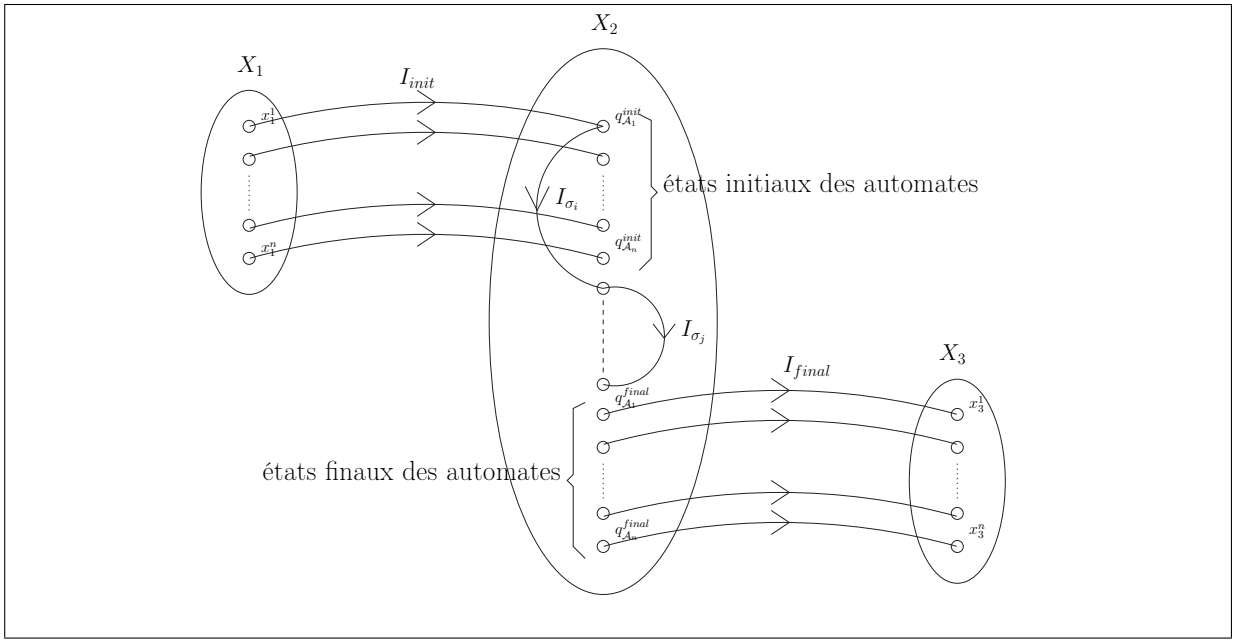


Fig. 8.16: La construction utilisée pour la preuve de complexité du problème MPIC

– I_{σ_i} $(1 \leq i \leq |\Sigma|)$ *de* X_2 *à* X_2 *:* $\forall k, 1 \leq k \leq |n|$ *et* $\forall l, 1 \leq l \leq |Q_{\mathcal{A}_k}|$, $I_{\sigma_i}\left(q^l_{\mathcal{A}_k}\right)$ *est défini si* $\delta_{\mathcal{A}_k}\left(q^l_{\mathcal{A}_k}, \sigma_i\right)$ *est défini et* $I_{\sigma_i}\left(q^l_{\mathcal{A}_k}\right) = \delta_{\mathcal{A}_k}\left(q^l_{\mathcal{A}_k}, \sigma_i\right)$

Si l'intersection des n automates injectifs est non nulle et que le mot $w = \sigma_s \dots \sigma_t$ *appartient à l'intersection (i.e. w est accepté par les n automates) alors la composition* $I_{comp} = I_{final} \circ I_{\sigma_t} \circ \dots I_{\sigma_s} \circ I_{init}$ *est une solution de l'instance du problème PIC construite, l'opposé est également vrai.* □

Proposition **2** *le problème MPIC est PSPACE-Difficile.*

Preuve **2** *Comme la version décisionnelle du problème d'optimisation MPIC est PSPACE-Complète, le problème MPIC est PSPACE-Difficile.* □

8.4.3 Algorithme pour la résolution du problème MPIC

L'algorithme proposé pour résoudre le problème MPIC ne se contente pas de trouver une solution mais fournit un moyen de connaître l'ensemble des solutions du problème MPIC. En effet, vue l'application qui est visée, c'est l'intégralité des solutions qui est potentiellement intéressante et pas seulement une seule solution.

Plutôt que d'énumérer l'ensemble des solutions du problème MPIC, l'algorithme proposé construit un automate qui reconnaît l'ensemble des injections partielles de taille maximale. L'avantage de construire un automate plutôt que d'énumérer directement les compositions vient du fait que le nombre de compositions peut croître exponentiellement avec le nombre d'états et de transitions de l'automate.

Avant de donner l'algorithme, il est nécessaire de poser quelques définitions préliminaires.

8.4.3.1 Définitions des automates AMPICAA et AMPICAAβ

Un automate reconnaissant toutes les solutions du problème MPIC peut être défini ainsi :

Définition **21** *Automate acceptant toutes les solutions du problème MPIC (AMPICAA)*
Soient $\mathcal{X} = \{X_1, \ldots, X_n\}$ un ensemble de n ensembles distincts, $\mathcal{I} = \{I_1, \ldots, I_m\}$ un ensemble de m injections partielles définies sur les ensembles de $\mathcal{X}$ et une paire d'ensembles $(X_i, X_j) \in \mathcal{X}^2$, l'automate d'états finis $\mathcal{A} = (Q, \mathcal{I}, \delta, q_{init}, F)$ est un AMPICAA (avec Q son ensemble d'états, $\mathcal{I}$ son ensemble fini de symboles, $\delta : Q \times \mathcal{I} \to Q$ sa fonction de transition, q_{init} son état initial et $F \subseteq Q$ son ensemble d'états finaux) de $(\mathcal{X}, \mathcal{I}, (X_i, X_j))$ si et seulement si :
$\forall I_{comp} \in \mathcal{I}^*$ $\quad$ *$\mathcal{A}$ accepte $I_{comp} \Leftrightarrow I_{comp}$ est une composition d'injections de taille maximale de X_i à X_j*

L'algorithme proposé construit un AMPICAA particulier nommé AMPICAAβ. Un AMPICAAβ est un AMPICAA où chaque état est, de plus, associé à un sous-ensemble d'un des ensembles de $\mathcal{X}$.

Définition **22** *AMPICAAβ*
Soient $\mathcal{A} = (Q, \mathcal{I}, \delta, q_{init}, F)$ un AMPICAA de $(\mathcal{X}, \mathcal{I}, (X_i, X_j))$, $\tilde{\mathcal{X}}$ l'ensemble de tous les sous-ensembles des X_i et une fonction injective $\beta : Q \to \tilde{\mathcal{X}}$ qui associe à chaque état de Q un élément unique de $\tilde{\mathcal{X}}$, on dit que $\mathcal{A}_\beta = \left(Q, \tilde{\mathcal{X}}, \mathcal{I}, \delta, \beta, q_{init}, F\right)$ est un AMPICAAβ de $(\mathcal{X}, \mathcal{I}, (X_i, X_j))$ si et seulement si :
- $\beta(q_{init}) = X_i$
- $\forall (q_i, q_k, I) \in (Q \times Q \times \mathcal{I}), \delta(q_i, I) = q_k \Rightarrow I(\beta(q_i)) = \beta(q_k)$

Note : la seconde condition garantit que l'ensemble associé à l'état q_i, successeur direct de l'état q_k, est égal à l'application de l'injection I, qui est associée à la transition entre les deux états, sur l'ensemble associé à l'état q_k.

8.4.3.2 Algorithme

L'algorithme construit l'automate AMPICAAβ en ajoutant des transitions et des états qui sont des successeurs des états créés précédemment. A l'initialisation, l'automate en cours de construction ne contient que l'état initial associé à l'intégralité de l'ensemble X_i. Afin d'atteindre le plus rapidement possible un état final associé à un sous-ensemble de

grande taille pour limiter la construction inutile d'états (dont la taille du sous-ensemble associé est inférieure à la taille des compositions d'injections partielles solutions), les états à partir desquels on ajoute des successeurs sont les états avec le plus grand sous-ensemble associé. Le premier état construit est l'état initial de l'automate associé à l'ensemble de départ.

Le pseudo-code de l'algorithme pour la construction de l'AMPICAAβ est donné à l'illustration algo 8.1.

Il est possible de modifier l'algorithme Meilleur-D-Abord pour que l'automate construit accepte l'ensemble des compositions d'injections partielles ayant une taille supérieure ou égale à un seuil n donné. Pour cela, il suffit de passer n en paramètre de l'algorithme, que la ligne 1 soit remplacée par $tailleMin \leftarrow n$ et que la ligne 17 soit supprimée.

8.4.4 Mise en œuvre

L'algorithme Meilleur-D-Abord a été implémenté en langage JAVA suivant le pseudo-code donné au paragraphe précédent.

```
Fonction MEILLEUR-D-ABORD → Automate
Paramètre : Ensemble : ensembleInitial,
            Ensemble : ensembleFinal,
            Ensemble-d-injections-partielles : I;
Variable : Automate : retour,
           Etat : nouvelEtat, etatCourant,
           Ensemble : ensembleCourant, nouvelEnsemble,
           Entier : tailleMin;
begin
1   tailleMin ← 1;
2   etatCourant ← créer un état;
3   associer etatCourant avec ensembleInitial et le ranger dans retour;
4   marquer etatCourant comme 'état initial';
5   tant que etatCourant ≠ null faire
6       ensembleCourant ← l'Ensemble associé à etatCourant;
7       pour chaque injection partielle i ∈ I applicable à ensembleCourant
        faire
8           nouvelEnsemble ← appliquer i à ensembleCourant ;
9           si |nouvelEnsemble| ≥ tailleMin alors
10              si il existe un état dans retour qui est déjà associé à nouvelEn-
                semble alors
11                  nouvelEtat ← rechercher l'état de retour associé à nouve-
                    lEnsemble;
12              sinon
13                  nouvelEtat ← créer un état;
14                  associer nouvelEtat avec nouvelEnsemble et le ranger dans
                    retour;
15                  si nouvelEnsemble ⊆ ensembleFinal alors
16                      marquer nouvelEtat comme 'état final';
17                      tailleMin ← max(tailleMin,
                                        tailleDe(nouvelEnsemble)) ;
18              créer une transition δ (etatCourant, i) = nouvelEtat;
19      marquer etatCourant comme 'exploré';
20      etatCourant ← rechercher l'état de retour non marqué 'exploré'
                      ou 'final' et associé à l'Ensemble de plus grande
                      taille (et de taille ≥ tailleMin) ;
21  enlever les états de retour associés à un Ensemble de taille inférieure stric-
    tement à tailleMin;
22  enlever les états et les transitions ne menant à aucun état 'final' de retour;
23  → retour;
end
```

ALG. **8.1:** L'algorithme MEILLEUR-D-ABORD pour la construction de l'automate acceptant toutes les compositions d'injections partielles de taille maximales entre deux ensembles

8.5 Expérimentations

Les expérimentations menées concernent la biosynthèse du tryptophane, l'un des trois acides aminés aromatiques, et la glycolyse (voir § 1.1.2.1 - page 9). Les trois acides aminés aromatiques (tryptophane, phénylalanine et Tyrosine) sont synthétisés à partir du même composé appelé chorismate, lui même dérivé de l'érythrose 4-phosphate et du PEP (voir § 1.1.2.2 - page 12) comme l'illustre la figure 8.17.

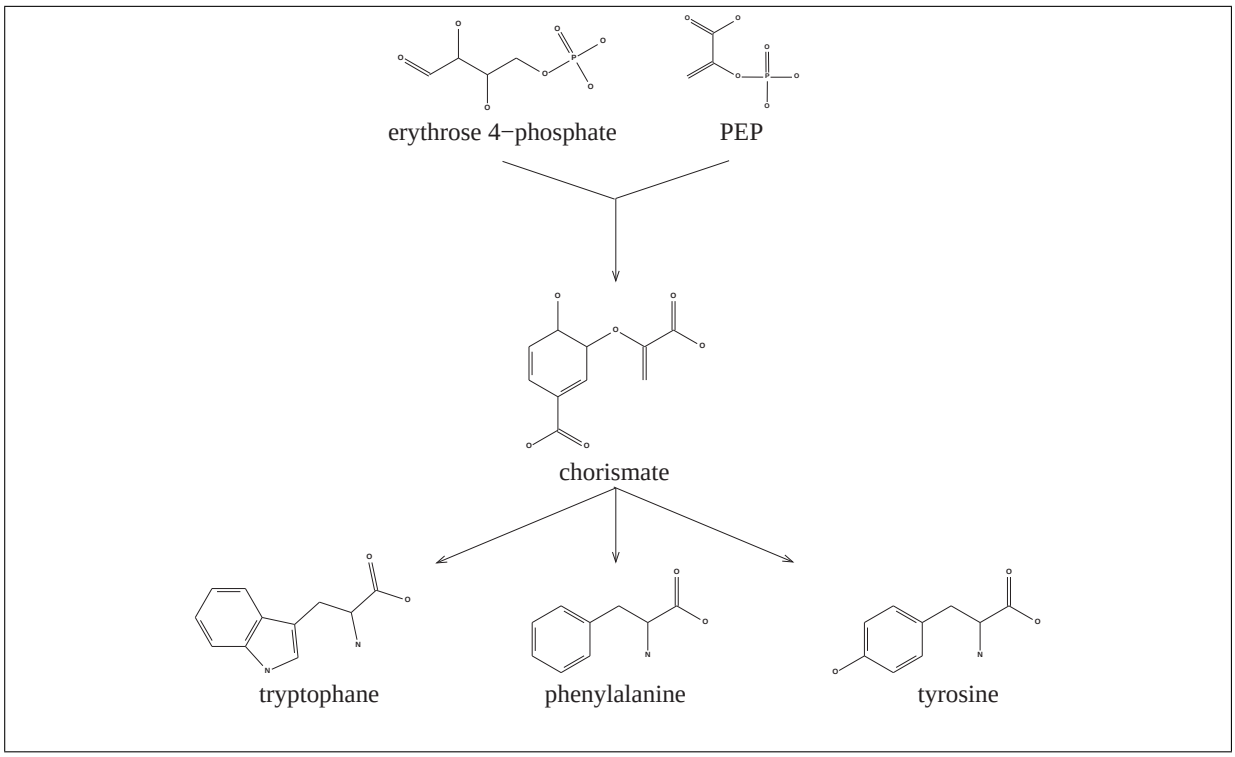


Fig. 8.17: La biosynthèse des acides aminés aromatiques se fait à partir du chorismate qui est synthétisé à partir d'érythrose 4-phosphate et de PEP

8.5.1 Post-traitement des résultats

L'algorithme proposé produit un automate reconnaissant toutes les compositions d'injections partielles de taille maximale (ou supérieure ou égale à un seuil fixé) entre deux ensembles.

Dans le contexte de la reconstruction de voies métaboliques, chaque chemin dans l'automate correspond à une succession de réactions entre les deux composés initial et final, mais certaines successions de réactions ne sont pas intéressantes. Il est possible de modifier l'automate en supprimant certains états et certaines transitions afin d'éviter ces successions particulières de réactions.

Dans le cas, illustré sur la figure 8.18, où un état D ne peut être atteint qu'à partir

d'un état B et où tous les chemins à partir de D vers un état final repassent par B, l'état D (et ceux, hormis B, qui lui sont liés) n'apporte rien d'intéressant et peut être supprimé.

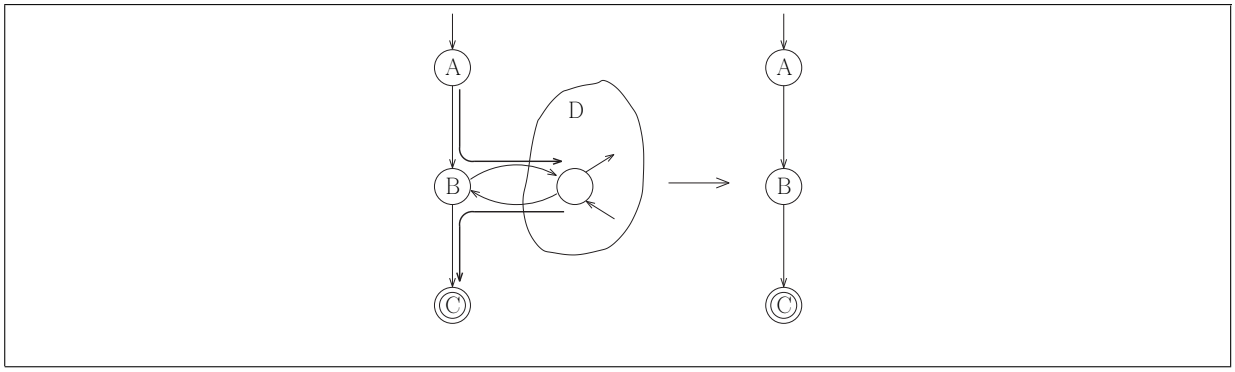


Fig. 8.18: Etats de l'automate AMPICAAβ pouvant être supprimés dans le contexte d'une application biologique

8.5.2 Données

Les données utilisées pour les expérimentations consistent en 6191 injections partielles différentes décrivant les transferts d'atomes de 3737 réactions impliquant 2920 composés différents. Toutes ces données ont été extraites de la banque LIGAND [Goto *et al.*, 2002] (version de novembre 2002). Pour l'application concernant la reconstruction de la voie de biosynthèse du tryptophane, les injections partielles ont été modifiées pour ne refléter que le transfert des atomes de carbone. Ainsi, le nombre d'injections partielles différentes est réduit à 4404, impliquant 2894 composés et 3721 réactions. De plus, toutes les réactions ont été considérées comme réversibles, aussi chaque injection partielle est définie deux fois (*i.e.* dans les deux directions de la réaction).

8.5.3 Reconstruction de la voie de biosynthèse du tryptophane

Pour la reconstruction de la voie de biosynthèse du tryptophane, trois automates différents ont été calculés :

- l'automate acceptant toutes les compositions de taille maximale allant de l'érythrose 4-phosphate au chorismate
- l'automate acceptant toutes les compositions de taille maximale allant du chorismate au tryptophane
- l'automate acceptant toutes les compositions de taille 6 allant du chorismate au tryptophane

Les propriétés de ces trois automates, avec les temps de construction, sont résumées dans la table 8.1. La construction de ces trois automates permet d'observer que la taille

des automates est d'autant plus grande que le nombre d'atomes transférés est faible. Cela vient du fait que le nombre d'injections à considérer est d'autant plus grand que la taille des injections peut être faible. Le premier automate (érythrose 4-phosphate vers chorismate) a des caractéristiques typiques d'un automate construit grâce à cette méthode, il comporte plusieurs centaines de nœuds et de transitions. On observe également que pour le troisième automate, reconnaissant les chemins allant du chorismate au tryptophane, le nombre d'états et de transitions a plus que quadruplé par rapport au second automate alors que la taille des injections résultantes acceptées par l'automate n'a diminuée que de 1 (de 7 à 6).

Enfin, on constate que les temps de calculs sont relativement courts et, en tout état de cause, sans comparaison avec ceux obtenus au chapitre 5.

Composé initial	Composé final	#carbones transférés	#états	#transitions	#injections ($\|\Sigma\|$)	temps de calcul
érythrose 4-phosphate	chorismate	4 (max)	5320	17427	1897	1'30"
chorismate	tryptophane	7 (max)	19	48	48	2"
chorismate	tryptophane	6	87	235	116	5"

Tab. 8.1: Caractéristiques des automates construits pour la voie de biosynthèse du tryptophane

La figure 8.19 montre l'automate acceptant tous les chemins réactionnels transférant au moins 6 atomes de carbone du chorismate au tryptophane. Les transitions de cet automate qui étaient utilisées uniquement dans la reconnaissance de chemins de longueur supérieure à 6 étapes ont été supprimées afin de limiter la taille de l'automate. Par soucis de clarté, dans cette figure, chaque état est représenté par la molécule complète à laquelle il est associé par la fonction β. Les transitions dessinées en gras correspondent aux réactions de la voie de biosynthèse connue. La voie de biosynthèse est complètement reconnue par l'automate. On peut également remarquer d'autres chemins différents dont un correspond à la voie de dégradation du tryptophane. La reconnaissance de cette voie par l'automate n'est pas surprenante étant donné que toutes les réactions ont été considérées comme réversibles.

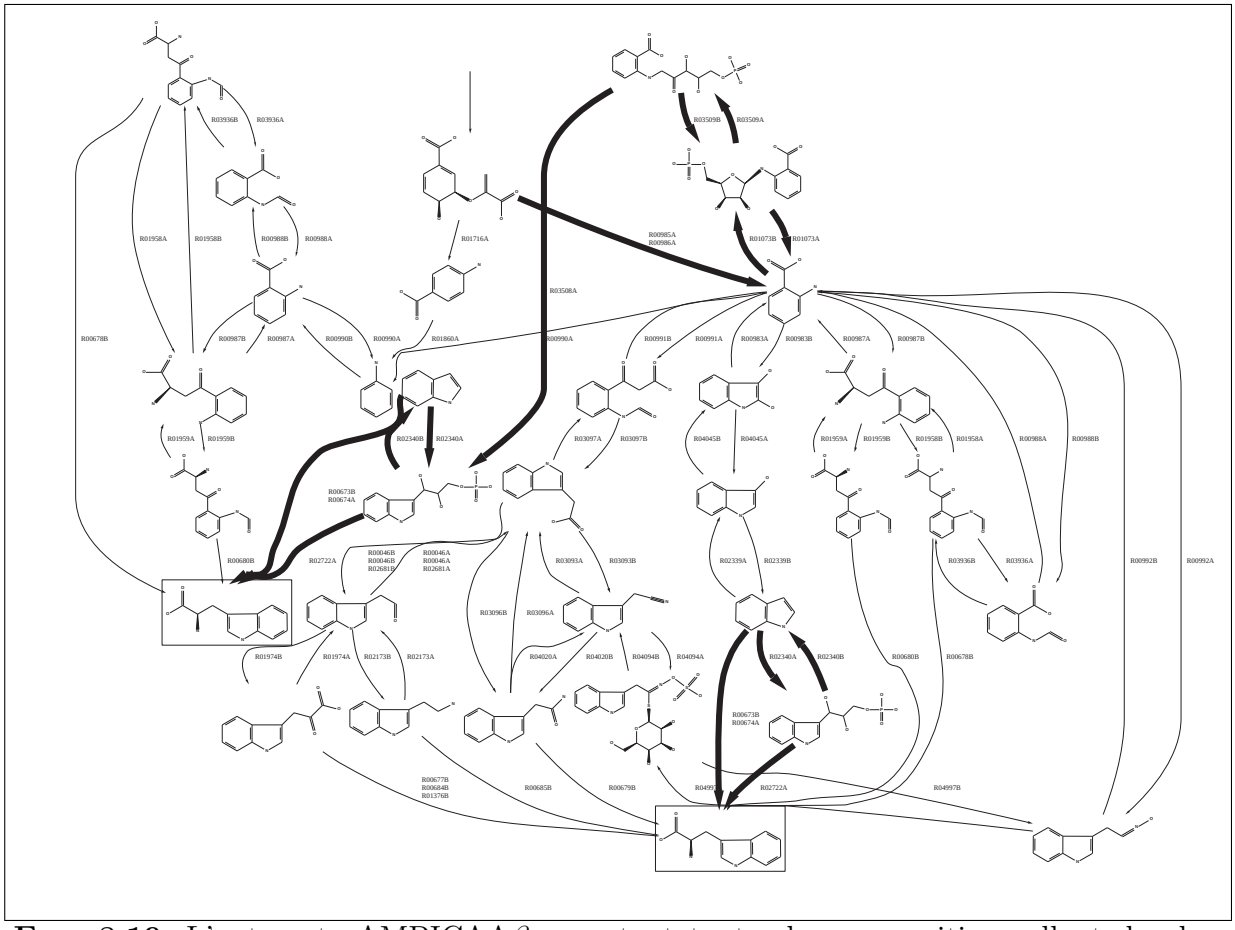


Fig. 8.19: L'automate AMPICAAβ acceptant toutes les compositions allant du chorismate au tryptophane transférant un minimum de 6 atomes de carbone et ayant une longueur maximale de 6 étapes. Les transitions de la voie de biosynthèse "de la littérature" sont indiquées en gras

8.5.4 Reconstruction de la glycolyse

Comme le montre l'exemple précédent, dans le cas où le nombre d'atomes à transférer est faible, le nombre d'états et de transitions des automates peut croître de façon dramatique. Aussi est-il dans certains cas utile de rechercher les chemins réactionnels qui transfèrent non pas uniquement les atomes de carbone mais tous les types d'atomes lourds (carbone, azote, oxygène, phosphore).

Afin de montrer le gain apporté par le choix des types d'atomes considérés, le cas de la glycolyse et plus particulièrement la sous-partie de la glycolyse faisant intervenir des composés phosphorylés (8 étapes sur les 10 de la glycolyse) peut être utilisé. Deux automates ont été calculés reconnaissant les chemins réactionnels correspondant à des injections dont la taille est maximale entre l'α-D-glucose 6-phosphate et le PEP. Le premier se contente de maximiser le nombre d'atomes de carbone transférés alors que le second maximise le nombre d'atomes lourds transférés.

Les propriétés de ces deux automates, avec les temps de construction, sont résumées dans la table 8.2. L'influence des types d'atome considérés est importante. Ainsi, l'automate reconnaissant les chemins réactionnels transférant 3 atomes de carbone a quasiment 30 fois plus d'états et de transitions que l'automate reconnaissant les chemins transférant 9 atomes lourds (C, N, O, P).

Composé initial	Composé final	#atomes transférés	#états	#transitions	#injections ($\|\Sigma\|$)	temps de calcul
érythrose 6-phosphate	PEP	3 carbones	20096	65324	2495	5'30"
érythrose 6-phosphate	PEP	9 atomes	685	2368	387	14'45"

Tab. 8.2: Caractéristiques des deux automates construits pour la glycolyse

La figure 8.20 montre la partie du second automate (acceptant les chemins transférant des atomes lourds) limitée aux transitions impliquées dans la reconnaissance de chemins de longueur 8 et inférieure. Dans cette figure, les transitions correspondant à des réactions impliquées dans la glycolyse sont dessinées en gras, on observe ainsi qu'elles forment un chemin dans l'automate. Cependant, une transition entre le fructose 1,6-diphosphate et le glycéraldéhyde 3-phosphate est manquante par rapport aux réactions connues de la glycolyse. Ceci est tout à fait normal car il n'y a qu'un seul groupement phosphate transféré du glucose 6-phosphate au fructose 1,6-diphosphate.

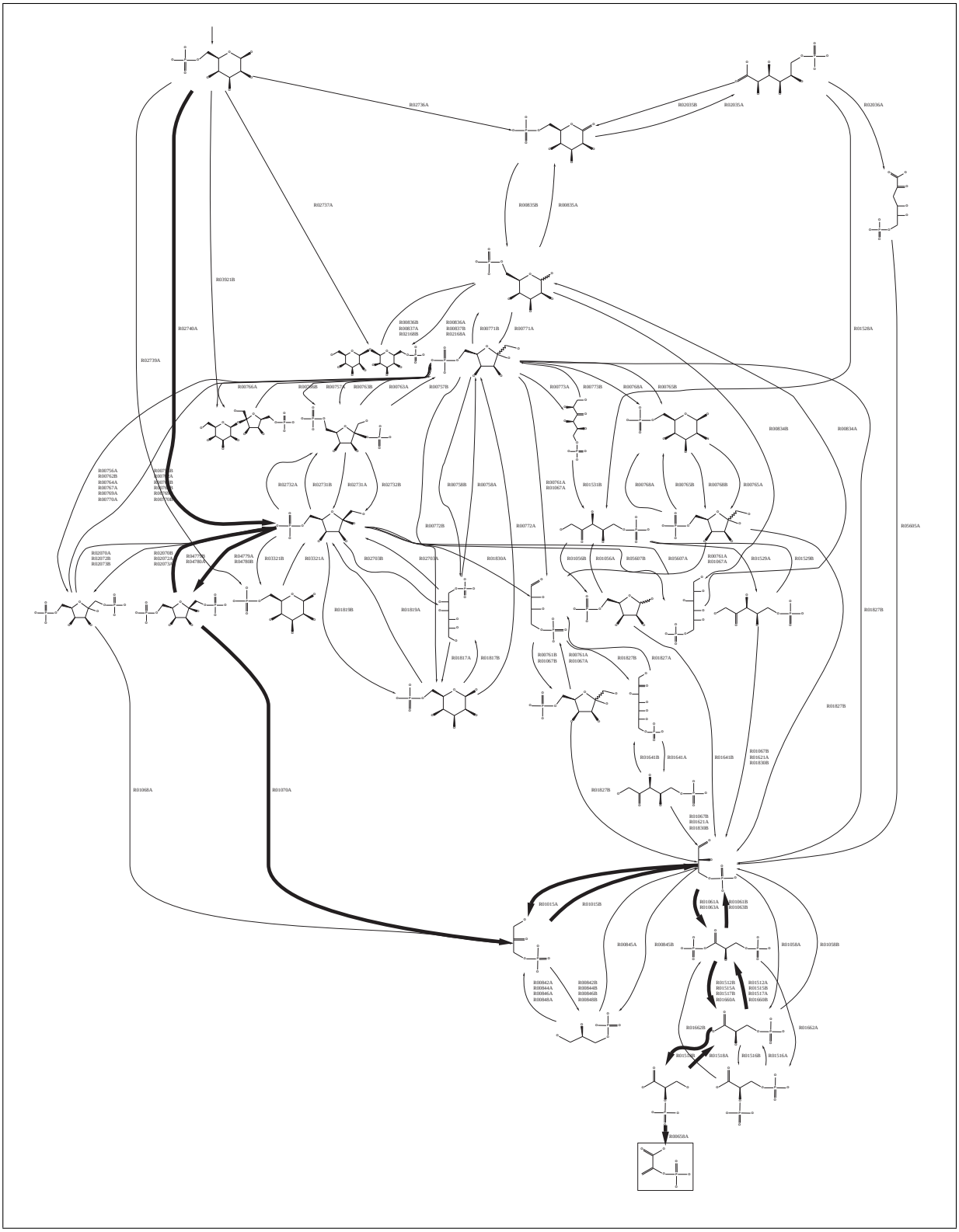


Fig. 8.20: L'automate AMPICAAβ acceptant toutes les compositions allant du glucose-6P au PEP transférant un minimum de 9 atomes lourds (C, N, O, P) et ayant une longueur maximale de 8 étapes. Les transitions correspondant à des réactions de la glycolyse sont indiquées en gras

8.6 Conclusion

A la vue des résultats obtenus sur la biosynthèse du tryptophane et de la glycolyse, cette approche semble être capable de reconstituer des voies métaboliques connues et cela sans autres connaissances *a priori* que les réactions elles-mêmes. Le seul paramètre nécessaire à l'approche est, le cas échéant, le nombre d'atomes à transférer entre le composé initial et le composé final, cela a l'avantage de permettre l'utilisation de l'outil de manière complètement exploratoire.

De plus, malgré une complexité théorique importante, le programme implémentant l'algorithme donné a des temps d'exécution qui permettent, en pratique, une utilisation interactive. La taille des réseaux retournés par la méthode n'est pas complètement rédhibitoire pour une analyse manuelle. De plus, il est tout à fait envisageable d'effectuer un traitement sur ces réseaux pour limiter le travail de l'expert biologiste en y incluant par exemple des informations sur la thermodynamique des réactions et/ou la connaissance d'un catalyseur biologique pour chaque réaction dans l'organisme d'intérêt.

Il faut cependant remarquer que cette méthode de reconstruction est dépendante des réactions qui lui sont données et surtout de la forme sous laquelle ces réactions sont données. Comme il a été montré au début de ce chapitre, cette partie peut cependant être traitée de façon automatique sur la plupart des réactions disponibles actuellement. Une limite de cette approche est qu'elle ne recherche que des chemins entre deux composés. Dans le cas d'une voie comme la biosynthèse du chorismate à partir de l'érythrose 4-phosphate et du PEP, il n'est pas possible d'inclure les deux substrats initiaux dans la requête. Il serait cependant tout à fait possible de traiter ce cas en comparant les réseaux obtenus pour les deux chemins (PEP vers chorismate et érythrose 4-phosphate vers chorismate) de façon adéquate. En effet, si il existe un réseau transformant du PEP et de l'érythrose 4-phosphate en chorismate, alors, les deux automates (PEP vers chorismate et érythrose 4-phosphate vers chorismate) reconnaissent tous les deux la partie inférieure de la voie. La partie inférieure de la voie correspond à un suffixe dans les mots reconnus par les automates. Il faut donc trouver les suffixes communs aux deux automates pour trouver la partie commune de la voie, le début du réseau étant donné par la façon, dans chaque automate, d'arriver jusqu'à l'état où commence la reconnaissance du suffixe commun.

Quatrième partie

Détermination de voies métaboliques procaryotes conservées codées en opérons

L'organisation des génomes en opérons est un caractère observé chez la plupart des organismes procaryotes (voir chapitre 6). La contiguïté des gènes codants pour des enzymes impliquées dans une même voie métabolique est donc une information importante qu'il peut être intéressant de mettre en jeu.

Suivant le point de vue ceci peut permettre :

i) de réduire les propositions de voies métaboliques (chapitre précédent) en s'appuyant sur l'organisation génique, ou

ii) de prédire des opérons en s'appuyant sur les données métaboliques

Nous décrivons dans ce chapitre une approche générale de la comparaison de graphes métaboliques et de graphes (d'intervalles) décrivant l'organisation génique.

Il faut noter que cette approche, fondée sur la recherche de composantes connexes communes à n graphes, est généralisable à d'autres applications biologiques (comme la comparaison de réseaux métaboliques avec des réseaux d'interaction protéine-protéine par exemple).

A titre d'application, nous traitons les prédictions pour l'ensemble des organismes bactériens connus afin d'identifier un ensemble de voies métaboliques codées en opérons conservées chez les organismes procaryotes.

Chapitre 9

Comparaison d'un réseau métabolique et de l'organisation des gènes sur le chromosome : extraction de voies métaboliques conservées codées en opérons dans les organismes procaryotes

9.1 Objectif

La co-régulation de gènes impliqués dans la même fonction est souvent prise en charge, dans les génomes bactériens par la structure en opéron. Aussi, il est attendu qu'une partie, variable selon les organismes, des enzymes impliquées dans une même voie métabolique soit codée par des gènes appartenant à un même opéron.

La présence au sein d'un même opéron de plusieurs gènes codant pour des enzymes impliquées dans une même voie métabolique permet d'affirmer que ces deux enzymes sont bien co-régulées (au moins au niveau transcriptionnel) et permet également d'avoir une plus grande confiance dans la fonction assignée à chacun des gènes. Enfin, la comparaison du nombre des opérons ainsi prédits, pour tous les organismes procaryotes entièrement séquencés, permet de mettre en évidence les organismes pour lesquels la régulation transcriptionnelle est largement prise en compte par les opérons et ceux pour lesquels d'autres moyens sont mis en œuvre.

9.2 Présentation du problème

Comme expliqué au § 6.2.2, un moyen de rechercher des opérons est de rechercher sur le chromosome un ensemble de gènes co-localisés et co-orientés codant pour des enzymes impliquées dans la même voie métabolique. Un exemple classique d'opéron très bien conservé chez les procaryotes est l'opéron contenant les gènes responsables de la voie de biosynthèse du tryptophane. La figure 9.1 montre les gènes responsables de la biosynthèse du tryptophane présents dans des opérons de trois bactéries (*Escherichia coli*, *Lactococcus lactis* et *Bacillus subtilis*) et de deux archæbactéries (*Archeoglobus fulgidus* et *Methanosarcina mazei*).

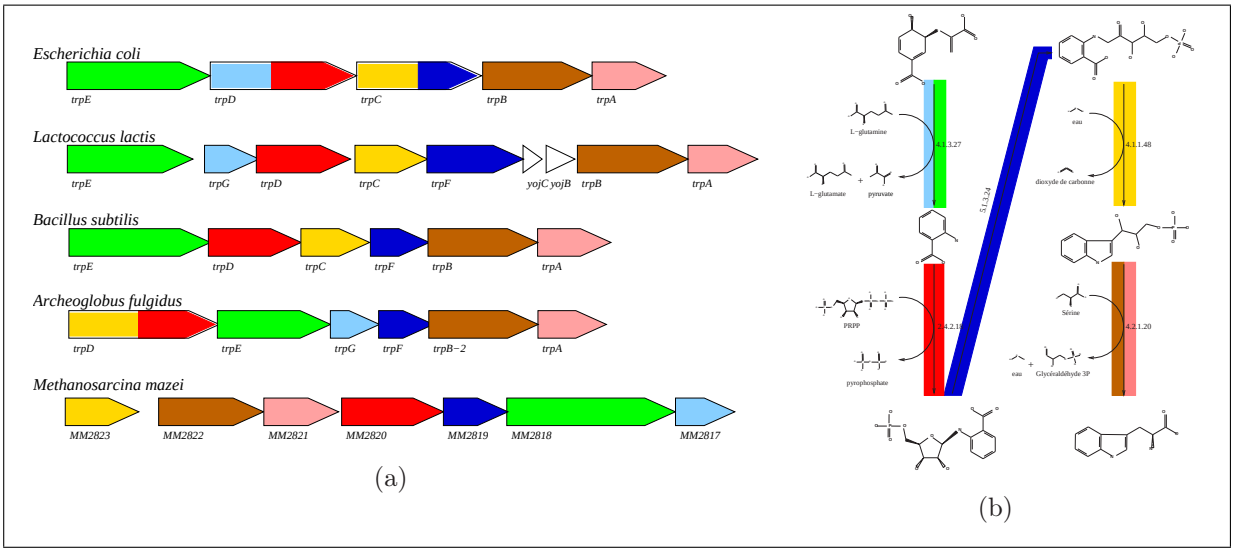


FIG. 9.1: Gènes en opérons responsables de la biosynthèse du tryptophane dans 3 espèces bactériennes et 2 espèces archæbactériennes - Les couleurs représentent des domaines fonctionnels impliqués dans la catalyse de certaines réactions. Les réactions catalysées par un complexe sont représentées par une flèche bicolore. On notera la présence de gènes qui ont fusionné chez *Escherichia coli* et *Archeoglobus fulgidus*, la disparition au sein de l'opéron de *Bacillus subtilis* d'un des gènes qui code pour une des sous-unités du complexe protéique impliqué dans la catalyse de la réaction d'EC 4.1.3.27, ainsi que l'insertion chez *Lactococcus lactis* de deux gènes au sein de l'opéron

Deux algorithmes ont été présentés au § 6.2.2 pour rechercher ce type d'opérons. Dans les deux travaux correspondants [Ogata *et al.*, 2000; Zheng *et al.*, 2002], le problème sous-jacent n'est pas formellement spécifié. Par ailleurs, les algorithmes présentés ne permettent pas toujours d'obtenir la solution exacte attendue dans des cas particuliers. Dans l'exemple simple de la figure 9.2, on s'attend à trouver 1 seul groupe de couples *(gène,réaction)* puisque tous ces couples sont à la fois connectés dans le premier et le second graphe. L'algorithme présenté au § 6.2.2.2 trouve dans ce cas 4 groupes séparés.

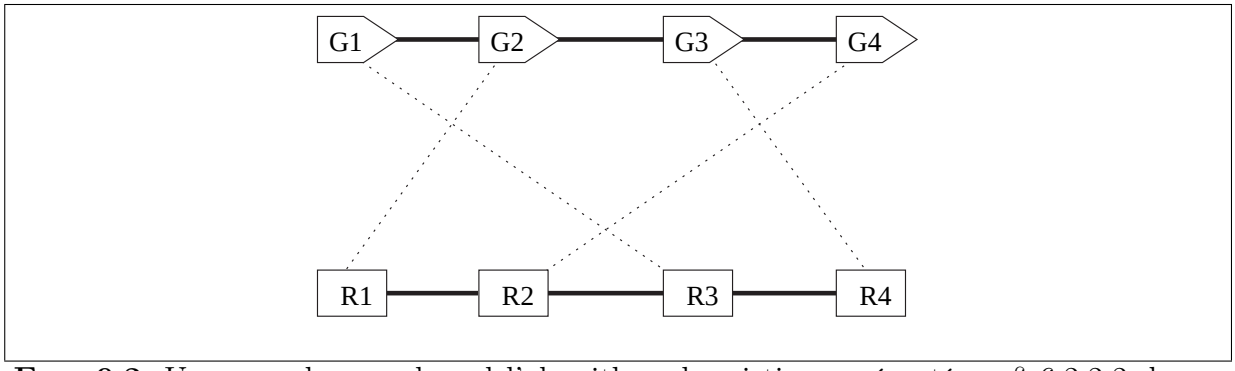


Fig. 9.2: Un exemple pour lequel l'algorithme heuristique présenté au § 6.2.2.2 donne un mauvais résultat

Cet exemple montre bien l'intérêt, d'une part, de formaliser ce problème et d'autre part de trouver un algorithme le résolvant exactement.

9.3 Formalisation du problème

9.3.1 Définition du problème strict

Dans l'exemple précédent, il est possible de représenter un génome et une voie métabolique sous la forme de graphes où, dans un cas, les nœuds représentent les gènes (Graphe Génome) et, dans le second, les réactions (Graphe Réactionnel) (figure 9.3). Le lien entre les deux graphes se fait au moyen des numéros EC, associés à la fois aux gènes et aux réactions.

On cherche donc à déterminer les ensembles maximaux de couples *(gène,réaction)* qui induisent dans chacun des 2 graphes des composantes connexes.

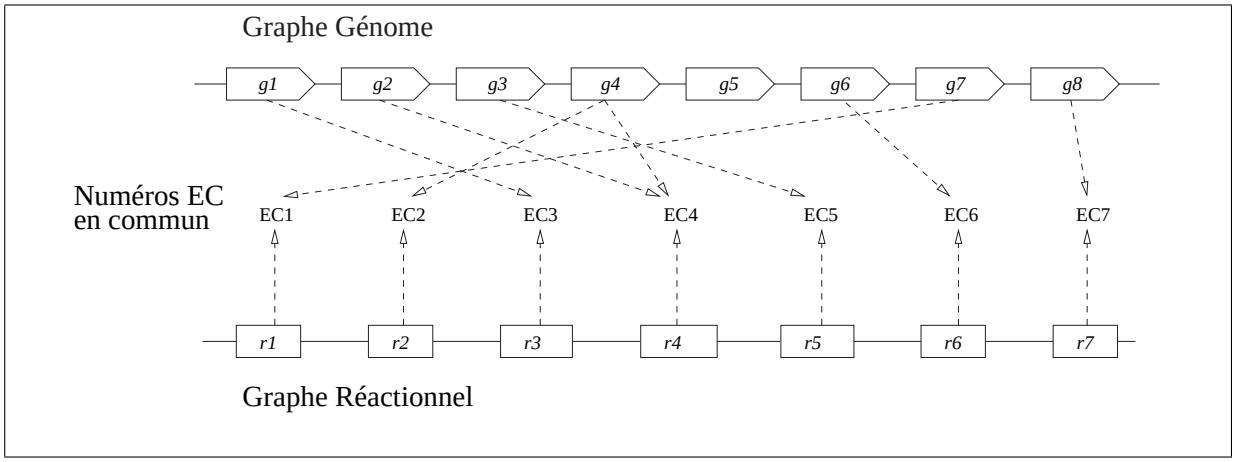


Fig. 9.3: Représentation du génome et du réseau métabolique par deux graphes - Les liens entre les nœuds sont les numéros EC, associés à la fois aux gènes et aux réactions

Dans ce but, définissons un graphe (appelé "multi-graphe de correspondance"), dont les nœuds sont des couples *(gène, réaction)* partageant le même numéro EC. Plus formellement, l'ensemble des nœuds est une restriction du produit cartésien des nœuds des deux graphes initiaux, la restriction est ici donnée par les numéros EC. Ces nœuds couples sont connectés par deux types d'arêtes, correspondant à chacun des deux graphes initiaux : deux nœuds couples sont reliés par une arête de type "génome" si les 2 gènes correspondant sont reliés dans le graphe Génome, de la même façon, deux nœuds couples sont reliés par une arête de type "réaction" si les 2 réactions correspondantes sont reliées dans le graphe Réactionnel.

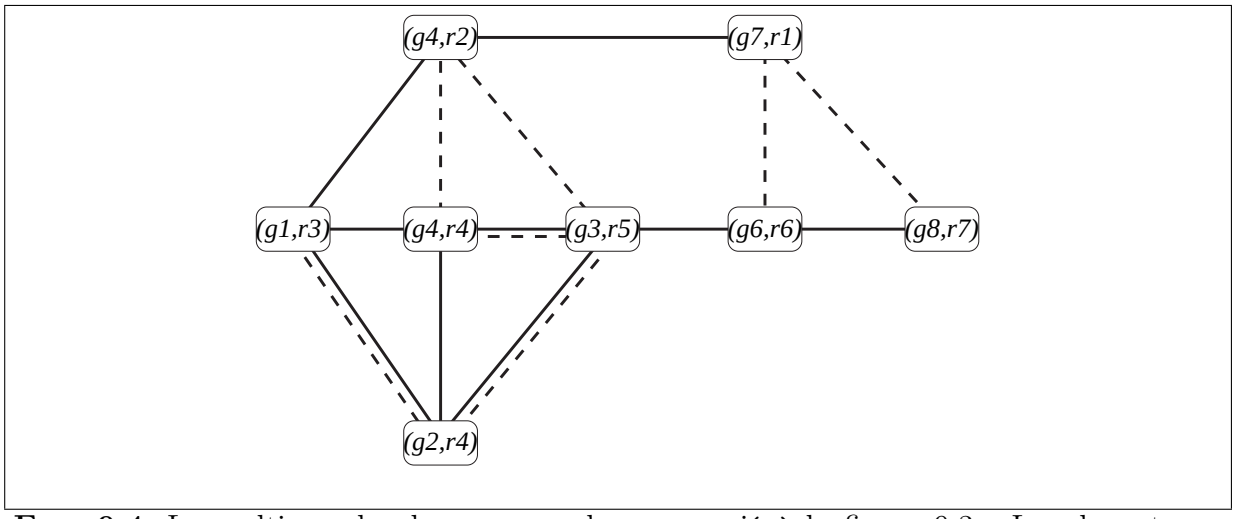


Fig. 9.4: Le multi-graphe de correspondance associé à la figure 9.3 - Les deux types d'arêtes, dessinés en traits pointillés et en traits pleins, correspondent respectivement au graphe Génome et au graphe Réactionnel. On remarque sur cette figure que, bien que le graphe Génome soit un graphe d'intervalles, le multigraphe est lui quelconque

Le problème initial de la recherche des opérons d'un organisme, à partir de l'organisation génomique des gènes et d'un réseau métabolique, s'exprime alors comme celui de la recherche des plus grandes composantes connexes communes à n graphes (représentés par le multi-graphe de correspondance). Nous présentons ici ce problème dans le cas général de $n \geq 2$ types d'arêtes différentes.

Problème **15** Composantes Connexes Communes Maximales (CCCMax)
Données : *un ensemble de n graphes $\{\mathcal{G}_1 = (V, E_1), \ldots, \mathcal{G}_n = (V, E_n)\}$ définis sur le même ensemble de nœuds V*
Réponse : *l'ensemble $\mathcal{X} = \{U_j \subseteq V\}$ tel que $\forall U_j \in \mathcal{X}$:*

- *U_j est une composante connexe de $\mathcal{G}_i, \forall i$*
- *U_j est de taille maximale*

Note importante : $\mathcal{X} = \{U_j\}$ forme une partition de V :

- tous les éléments de V appartiennent à un U_j
- $\forall j, k, j \neq k, U_j \cap U_k = \emptyset$ (si U_j et U_k ont une intersection non vide alors U_j et U_k ne sont pas maximaux car, dans ce cas précis, $U_j \cup U_k$ est une composante connexe pour tous les $\mathcal{G}_i$)

Remarque : par construction, chaque ensemble de nœuds associé à un élément de la partition forme une composante connexe dans tous les graphes initiaux.

Exemple : la figure 9.5 montre la partition solution du problème CCCMax sur l'exemple de la figure 9.4.

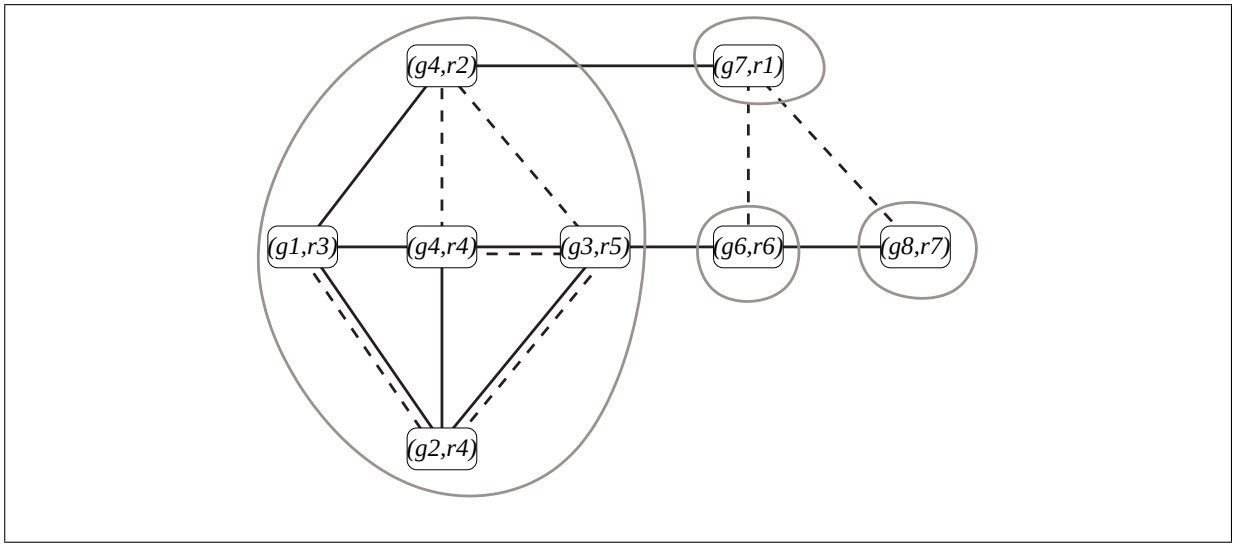


Fig. 9.5: Exemple d'instance du problème CCCMax - La partition résultat est de taille 4. L'ensemble des nœuds $\{(g6, r6), (g7, r1), (g8, r7)\}$ du multi-graphe de correspondance ne fait pas partie de la solution car, $\{(g6, r6), (g7, r1), (g8, r7)\}$ n'est pas une composante connexe du graphe Réactionnel

Ce problème a initialement été formulé ainsi par [Morgat, 2001] pour l'étude des micro-synténies bactériennes. Il s'agissait, dans ce cas, de comparer les deux graphes (d'intervalles) associés aux génomes de deux organismes (graphe "Génome" de la figure 9.3). Nous l'étendons ici au cas des graphes métaboliques.

9.3.1.1 Relaxation de la contrainte de contiguïté stricte

La définition du problème 15 contraint les nœuds de chaque élément de la partition $\mathcal{X}$ à induire une composante connexe dans les graphes initiaux, ce qui est une contrainte forte. Il est possible de relaxer cette contrainte en autorisant des "gaps" dans les graphes initiaux. Une façon simple d'obtenir ceci sans changer la définition du problème consiste à "prétraiter" les graphes initiaux en les fermant partiellement.

DÉFINITION **23** *Graphe δ-fermé*
Soit un graphe $\mathcal{G} = (V, E)$ et un entier positif δ, le graphe δ-fermé, noté $\mathcal{G}^\delta = \left(V, E^\delta\right)$, est un graphe défini sur le même ensemble de nœuds que $\mathcal{G}$. Son ensemble d'arêtes E^δ est défini tel que :

- $(u, v) \in E^\delta \Leftrightarrow$ *il existe un chemin de u à v dans $\mathcal{G}$ de longueur inférieure ou égal à* $(1 + \delta)$

Pour obtenir le graphe δ-fermé à partir d'un graphe, on peut utiliser un algorithme comme celui de Floyd [Floyd, 1962].

Exemple : la figure 9.6 montre le multi-graphe obtenu à partir des graphes de la figure 9.3 si le paramètre δ, pour le premier graphe, est fixé à 1. Dans ce cas, la partition solution est l'ensemble de tous les nœuds du multi-graphe car ils forment bien une composante connexe du graphe du génome, fermé partiellement avec $\delta = 1$, et du second graphe.

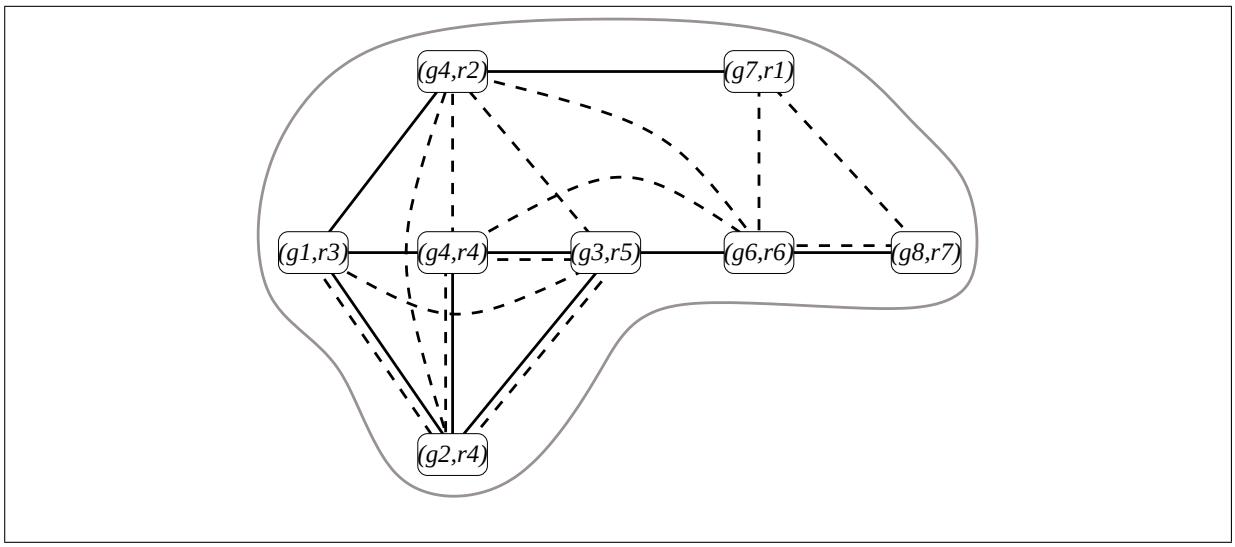


FIG. 9.6: Exemple de relaxation de la contrainte de contiguïté stricte pour le problème CCCMAX - Avec $\delta = 1$ pour le graphe du génome (arêtes en pointillés) la partition solution du problème CCCMAX n'a plus qu'un seul élément qui contient tous les nœuds du multi-graphe de correspondance

9.4 Algorithme et complexité

9.4.1 Algorithme

L'algorithme que nous proposons pour résoudre le problème CCCMAX consiste à construire la partition $\mathcal{X}$ par raffinements successifs à partir d'une grande classe initiale contenant tous les nœuds de V. A chaque étape, on raffine les classes de $\mathcal{X}$ qui ne satisfont

pas à la définition (c'est-à-dire qu'il existe au moins un $\mathcal{G}_i$ tel que la classe n'induit pas une composante connexe). Ces classes sont dites instables. L'algorithme s'arrête lorsque toutes les classes sont stables. Etant donnée une classe U_j instable (c'est-à-dire qu'il existe au moins un $\mathcal{G}_i$ tel que U_j n'induit pas une composante connexe), cette classe doit être éclatée en autant de classes qu'il y a d'intersections non vides de composantes connexes des $\mathcal{G}_i$. Cette condition est nécessaire mais non suffisante car il est possible que les classes résultantes soient instables. C'est pourquoi ces classes doivent à nouveau être raffinées.

Ceci est illustré sur la figure 9.7 où la partition solution ne correspond pas à l'intersection des composantes connexes et met en exergue le fait que le calcul de la partition demande plusieurs calculs successifs d'intersections de composantes connexes. Pour cet exemple, la première intersection des composantes connexes donne $\{\{a\},\{b,c\}\}$. La classe $\{b,c\}$ est instable et doit être raffinée. La partition finale correcte est $\{\{a\},\{b\},\{c\}\}$.

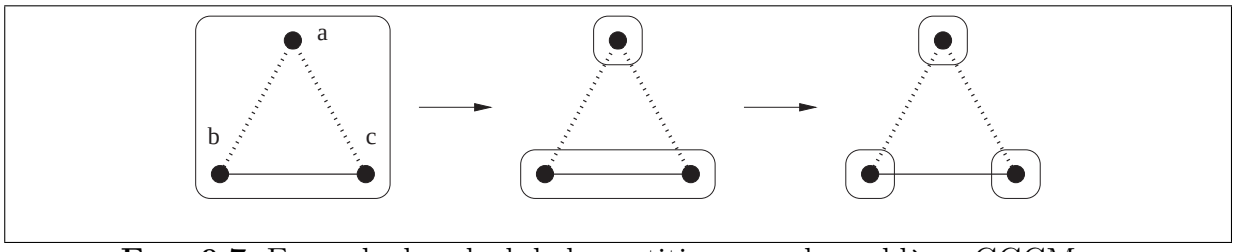


Fig. 9.7: Exemple de calcul de la partition pour le problème CCCMax

Le pseudo-code de l'algorithme est donné sans le détail du calcul des intersections ni du calcul des composantes connexes et présente une version récursive de l'algorithme (voir algorithme 9.1).

9.4.2 Complexité

9.4.2.1 Construction du multi-graphe de correspondance

La taille du multi-graphe de correspondance, c'est-à-dire son nombre de nœuds dépend de la fonction de restriction. Dans le pire des cas, chaque nœud de chacun des n graphes initiaux est compatible avec les nœuds de tous les autres graphes. Dans ce cas, si chacun n graphes a de l'ordre de M nœuds, alors, le multi-graphe de correspondance construit à $\mathcal{O}(M^n)$ nœuds. La phase de prétraitement, pour l'ajout des arêtes (pour la fermeture partielle), est effectuée en temps $\mathcal{O}(M^3)$ pour chacun des n graphes.

9.4.2.2 Construction de la partition

Ainsi que le pseudo-code de l'algorithme le montre, la construction de la partition s'effectue en 2 opérations successives. Dans un premier temps, on calcule les composantes connexes, ensuite, on calcule leurs intersections.

```
Fonction PartitionCCCMax → Ensemble-d-ensembles-de-noeuds
Paramètre : Ensemble-de-noeuds : V,
            Ensemble-d-ensembles-d-arêtes : E
Variable : Tableau-d-ensembles-d-ensembles-de-noeuds : composantes,
           Ensemble-d-ensembles-de-noeuds : lesClasses,
           Ensemble-de-noeuds : uneClasse,
           Ensemble-d-ensembles-de-noeuds : laPartition
début
    pour chaque i ∈ [1..|E|] faire
        composantes[i] ← composantesConnexes (G = (V, E_i));
    lesClasses ← intersectionComposantes (composantes);
    si |lesClasses| = 1 alors
        /*
        V est stable,
        l'ensemble des nœuds est une composante connexe pour
        chaque E_i
        */
        laPartition ← lesClasses;
    sinon
        /*
        V est instable,
        il faut refaire des appels récursifs pour chaque intersection et
        composer les résultats
        */
        laPartition ← ∅ ;
        pour chaque uneClasse ∈ lesClasses faire
            laPartition ← laPartition ∪
                          PartitionCCCMax (uneClasse, E);
    → laPartition;
fin

Programme
Paramètre : Multi-Graphe-de-correspondance : G = (V, {E_i}) ;
début
    PartitionCCCMax(V, {E_i});
fin Programme
```

Alg. **9.1:** Pseudo-code pour la résolution du problème CCCMax

Composantes connexes Pour chaque classe instable $C \subseteq V$, il faut calculer les composantes connexes de chacun des n sous-graphes induits $\mathcal{G}_i^C = \left(C, E_i^C = E_i \cap C^2\right)$. Pour une classe C et un sous-graphe, ce calcul nécessite $\mathcal{O}\left(|C| + \left|E_i^C\right|\right)$ opérations. Donc, pour toutes les classes de la partition, ceci est borné supérieurement par $\mathcal{O}\left(|V| + |E_i|\right)$ (car $\sum^{C} \left|E_i^C\right| \leq |E_i|$).

En sommant finalement sur tous les n graphes $\mathcal{G}_i$, on obtient un coût final de $\mathcal{O}\left(n \times |V| + \sum_i^n |E_i|\right)$.

Intersections des composantes connexes Pour chacune des classes instables $C \subseteq V$, il faut ensuite calculer l'intersection des composantes connexes. En utilisant un tri par casier [Karp *et al.*, 1972], on peut effectuer cette intersection en $\mathcal{O}\left(|C| \times (n-1)\right)$. Pour l'ensemble des classes, le coût total est donc de $\mathcal{O}\left(|V| \times (n-1)\right)$.

Chaque étape de l'algorithme a ainsi un coût de $\mathcal{O}\left(n \times |V| + \sum_i^n |E_i| + |V| \times (n-1)\right)$.

Comme une partition a un maximum de $|V|$ éléments et que dans le pire cas on construit une classe par étape, il faut effectuer au plus $|V|$ étapes. Ce qui finalement donne une complexité de $\mathcal{O}\left(|V| \times \left(n \times |V| + \sum_i^n |E_i| + |V| \times (n-1)\right)\right)$

soit $\mathcal{O}\left(|V| \times \left(n\,|V| + \sum_i^n |E_i|\right)\right)$

En pratique, le nombre d'étapes qu'il faut effectuer est souvent très largement inférieur à $|V|$. Dans tous les tests effectués, nous n'avons jamais dépassé les 10 étapes pour des graphes de plusieurs milliers de nœuds. De plus sauf pour des cas pathologiques (que l'on peut construire), le nombre d'étapes croît très faiblement avec $|V|$ ce qui rend l'algorithme pratiquement linéaire avec le nombre de nœuds.

Note : Un algorithme de complexité moindre ($\mathcal{O}\left(|V|\,.\log\left(|V|\right) + \sum_i^n |E_i|.\log^2\left(|V|\right)\right)$) se basant sur des structures de données complexes est décrit dans [Habib *et al.*, 2003]. Nous n'avons pu comparer le comportement en pratique des deux algorithmes, aucune implémentation de cet algorithme étant disponible.

9.5 Applications

L'algorithme de partition présenté précédemment a été implémenté en langage C. Nous l'avons appliqué à la prédiction des opérons codant pour des voies métaboliques pour tous les génomes bactériens et archæbactériens complètement séquencés.

Ces résultats sont ensuite réutilisés afin de rechercher des opérons conservés dans les γ-protéobactéries.

9.5.1 Données

Pour appliquer le problème CCCMax à la recherche des opérons correspondant à des voies métaboliques dans un organisme, les données du problème sont :

- un graphe circulaire $\mathcal{G}_{\text{génome}} = (V_g, E_g)$ représentant le génome de l'organisme étudié. Dans ce graphe, chaque nœud représente un gène, deux nœuds sont reliés par une arête si les deux gènes sont voisins et co-orientés sur le chromosome.
- un graphe $\mathcal{G}_{\text{métabolique}} = (V_m, E_m)$ représentant le réseau métabolique (global ou restreint aux réactions catalysées dans l'organisme d'intérêt). Dans ce graphe chaque nœud correspond à une réaction. Deux nœuds sont reliés par une arête si les deux réactions partagent un substrat ou un produit en commun (graphe $\mathcal{G}_R$ du § 3.1, page 29).

La correspondance entre gènes et réactions est donnée par les numéros EC.

9.5.1.1 Génomes

Les génomes utilisés sont extraits des fichiers génomes de l'EBI [EBI, 2004]. Les liens entre numéros EC et gènes ont été extraits des fichiers de l'EBI, des fiches HAMAP [Gattiker *et al.*, 2003] et également prédits à partir de la banque ENZYME [Bairoch, 2000] suivant la procédure décrite dans [Renard-Claudel *et al.*, 2003]. Les 153 génomes disponibles en mars 2004 ont été utilisés. Le tableau 9.1 recense l'intégralité des génomes avec les mnémoniques associés utilisés dans les illustrations qui suivent.

9.5.1.2 Graphe métabolique

Le graphe des réactions a été construit à partir de la banque LIGAND/KEGG [Goto *et al.*, 2002] suivant la méthode présentée au § 3.1. Un exemple de graphe de réaction est donné sur la figure 3.1(c). Pour les liens entre les réactions, seuls les métabolites contenant au moins 2 atomes de carbone ont été considérés (afin d'éliminer les petits composés comme l'eau, le dioxyde de carbone, l'ammoniac, le pyrophosphate) ainsi que les composés intervenant dans moins de 220 réactions (afin d'éliminer ATP, ADP, NAD^+, $NADP^+$, NADH et NADPH).

Mnemo	Nom	Mnemo	Nom	Mnemo	Nom
aeperAa	Aeropyrum pernix	agtumAa	Agrobacterium tumefaciens str. C58 (Cereon)	agtumBa	Agrobacterium tumefaciens str. C58 (U. Washington)
aqaeoAa	Aquifex aeolicus VF5	arfulAa	Archaeoglobus fulgidus DSM 4304	baantAa	Bacillus anthracis str. Ames
bacerAa	Bacillus cereus ATCC 14579	bahalAa	Bacillus halodurans	basubAa	Bacillus subtilis subsp. subtilis str. 168
batheAa	Bacteroides thetaiotaomicron VPI-5482	bdbacAa	Bdellovibrio bacteriovorus	bilonAa	Bifidobacterium longum NCC2705
bobroAa	Bordetella bronchiseptica	boburAa	Borrelia burgdorferi B31	boparAa	Bordetella parapertussis
boperAa	Bordetella pertussis	brjapAa	Bradyrhizobium japonicum USDA 110	brmelAa	Brucella melitensis 16M
brmelBa	Brucella suis 1330	buaphAa	Buchnera aphidicola str. APS (Acyrthosiphon pisum)	buaphBa	Buchnera aphidicola str. Bp (Baizongia pistaciae)
buaphCa	Buchnera aphidicola str. Sg (Schizaphis graminum)	cabloAa	Candidatus Blochmannia floridanus	cajejAa	Campylobacter jejuni subsp. jejuni NCTC 11168
cavibAa	Caulobacter crescentus CB15	chcavAa	Chlamydophila caviae GPIC	chmurAa	Chlamydia muridarum
chpneAa	Chlamydophila pneumoniae AR39	chpneBa	Chlamydophila pneumoniae CWL029	chpneCa	Chlamydophila pneumoniae TW-183
chpneDa	Chlamydophila pneumoniae J138	chtepAa	Chlorobium tepidum TLS	chtraAa	Chlamydia trachomatis
chvioAa	Chromobacterium violaceum ATCC 12472	claceAa	Clostridium acetobutylicum	clperAa	Clostridium perfringens str. 13
cltetAa	Clostridium tetani E88	coburAa	Coxiella burnetii RSA 493	coeffAa	Corynebacterium efficiens YS-314
cogluAa	Corynebacterium glutamicum ATCC 13032	cogluBa	Corynebacterium glutamicum ATCC 13032	deradAa	Deinococcus radiodurans
enfaeAa	Enterococcus faecalis V583	escolAa	Escherichia coli K12	escolBa	Escherichia coli O157 :H7 EDL933
escolCa	Escherichia coli O157 :H7	escolDa	Escherichia coli CFT073	funucAa	Fusobacterium nucleatum subsp. nucleatum ATCC 25586
gesulAa	Geobacter sulfurreducens PCA	glvioAa	Gloeobacter violaceus	haducAa	Hæmophilus ducreyi 35000HP
hainfAa	Hæmophilus influenzae Rd KW20	hasppAa	Halobacterium sp. NRC-1	hehepAa	Helicobacter hepaticus ATCC 51449
hepylAa	Helicobacter pylori 26695	hepylBa	Helicobacter pylori J99	lajohAa	Lactobacillus johnsonii
lalacAa	Lactococcus lactis subsp. lactis	laplaAa	Lactobacillus plantarum WCFS1	leintAa	Leptospira interrogans serovar lai str. 56601
liinnAa	Listeria innocua	limonAa	Listeria monocytogenes	meaceAa	Methanosarcina acetivorans C2A
mejanAa	Methanocaldococcus jannaschii	mekanAa	Methanopyrus kandleri AV19	melotAa	Mesorhizobium loti
memarAa	Methanococcus maripaludis	memazAa	Methanosarcina mazei Goe1	metheAa	Methanothermobacter thermautotrophicus str. Delta H
myaviAa	Mycobacterium avium subsp. paratuberculosis	mybovAa	Mycobacterium bovis subsp. bovis AF2122/97	mygalAa	Mycoplasma gallisepticum R
mygenAa	Mycoplasma genitalium	mylepAa	Mycobacterium leprae	mymycAa	Mycoplasma mycoides subsp. mycoides SC
mypenAa	Mycoplasma penetrans	mypneAa	Mycoplasma pneumoniae	mypulAa	Mycoplasma pulmonis
mytubAa	Mycobacterium tuberculosis H37Rv	mytubBa	Mycobacterium tuberculosis CDC1551	nemenAa	Neisseria meningitidis Z2491
nemenBa	Neisseria meningitidis MC58	nieurAa	Nitrosomonas europaea ATCC 19718	nosppAa	Nostoc sp. PCC 7120
ociheAa	Oceanobacillus iheyensis	pamulAa	Pasteurella multocida	phlumAa	Photorhabdus luminescens subsp. laumondii TTO1
pisppAa	Pirellula sp.	poginAa	Porphyromonas gingivalis W83	prmarAa	Prochlorococcus marinus subsp. marinus str. CCMP1375
prmarBa	Prochlorococcus marinus subsp. pastoris str. CCMP1986	prmarCa	Prochlorococcus marinus str. MIT 9313	psaerAa	Pseudomonas aeruginosa PAO1
psputAa	Pseudomonas putida KT2440	pssyrAa	Pseudomonas syringae pv. tomato str. DC3000	pyabyAa	Pyrococcus abyssi
pyaerAa	Pyrobaculum aerophilum	pyfurAa	Pyrococcus furiosus DSM 3638	pyhorAa	Pyrococcus horikoshii
rasolAa	Ralstonia solanacearum	rhpalAa	Rhodopseudomonas palustris	rhpalBa	Rhodopseudomonas palustris
riconAa	Rickettsia conorii	riproAa	Rickettsia prowazekii	saentAa	Salmonella enterica subsp. enterica serovar Typhi
saentBa	Salmonella enterica subsp. enterica serovar Typhi Ty2	satypAa	Salmonella typhimurium LT2	shfleAa	Shigella flexneri 2a str. 301
shfleBa	Shigella flexneri 2a str. 2457T	shoneAa	Shewanella oneidensis MR-1	simelAa	Sinorhizobium meliloti
stagaAa	Streptococcus agalactiae 2603V/R	stagaBa	Streptococcus agalactiae NEM316	staurAa	Staphylococcus aureus subsp. aureus N315
staurBa	Staphylococcus aureus subsp. aureus Mu50	staurCa	Staphylococcus aureus subsp. aureus MW2	staveAa	Streptomyces avermitilis
stcoeAa	Streptomyces coelicolor	stepiAa	Staphylococcus epidermidis ATCC 12228	stmutAa	Streptococcus mutans UA159
stpneAa	Streptococcus pneumoniae TIGR4	stpneBa	Streptococcus pneumoniae R6	stpyoAa	Streptococcus pyogenes M1 GAS
stpyoBa	Streptococcus pyogenes MGAS8232	stpyoCa	Streptococcus pyogenes SSI-1	stpyoDa	Streptococcus pyogenes MGAS315
susolAa	Sulfolobus solfataricus	sutokAa	Sulfolobus tokodaii	sysppAa	Synechocystis sp. PCC 6803
syspqAa	Synechococcus sp. WH 8102	thaciAa	Thermoplasma acidophilum	theloAa	Thermosynechococcus elongatus BP-1
thmarAa	Thermotoga maritima	thtenAa	Thermoanaerobacter tengcongensis	thvolAa	Thermoplasma volcanium
trdenAa	Treponema denticola ATCC 35405	trpalAa	Treponema pallidum subsp. pallidum str. Nichols	trwhiAa	Tropheryma whipplei str. Twist
trwhiBa	Tropheryma whipplei TW08/27	ururAa	Ureaplasma urealyticum	vichoAa	Vibrio cholerae O1 biovar eltor str. N16961
viparAa	Vibrio parahaemolyticus	vivulAa	Vibrio vulnificus CMCP6	vivulBa	Vibrio vulnificus YJ016
wigloAa	Wigglesworthia glossinidia endosymbiont of Glossina brevipalpis	woendAa	Wolbachia endosymbiont of Drosophila melanogaster	wosucAa	Wolinella succinogenes
xaaxoAa	Xanthomonas axonopodis pv. citri str. 306	xacamAa	Xanthomonas campestris pv. campestris str. ATCC 33913	xyfasAa	Xylella fastidiosa 9a5c
xyfasBa	Xylella fastidiosa Temecula1	yepesAa	Yersinia pestis CO92	yepesBa	Yersinia pestis KIM

Tab. 9.1: L'ensemble des génomes utilisés pour les expérimentations

9.5.2 Application à la recherche de voies métaboliques codées en opérons chez *Escherichia coli* : influence du paramètre $\delta_{\text{génome}}$

Afin de mesurer l'influence du paramètre $\delta_{\text{génome}}$, paramètre qui relâche la contrainte de contiguïté sur le chromosome, les opérons ont été recherchés avec $\delta_{\text{génome}}$ allant de 0 à 5 pour *Escherichia coli (K12)*. La figure 9.8 montre les distributions des tailles des opérons pour chacune de ces valeurs.

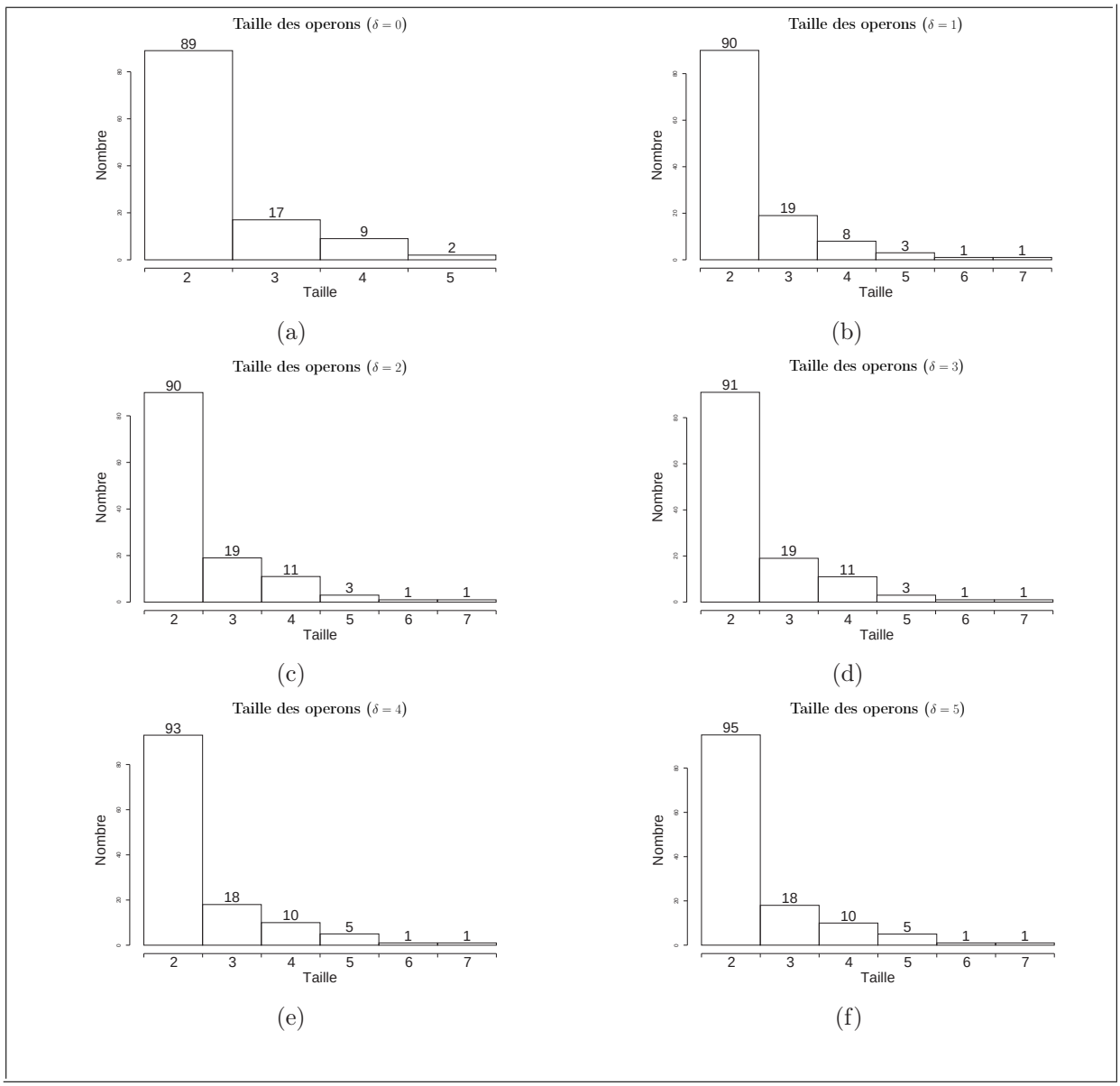


Fig. 9.8: Impact du paramètre $\delta_{\text{génome}}$ sur la taille des opérons prédits pour *Escherichia coli (K12)* - Seuls les gènes impliqués dans la catalyse des réactions du réseau réactionnel associé sont pris en compte

On remarque que les distributions sont peu différentes et que l'impact du paramètre

$\delta_{\text{génome}}$ est donc limité.

Il est à noter que l'opéron de plus grande taille trouvé pour ce génome (trouvé avec $\delta \geq 1$) n'est autre que l'opéron codant pour une partie de la biosynthèse du peptidoglycane montré sur la figure 6.6 (page 101).

9.5.3 Application à la recherche de voies métaboliques codées en opérons conservés dans les génomes bactériens et archæbactériens complets

Pour chaque organisme, la procédure décrite au dessus a été appliquée avec comme valeurs des paramètres $\delta_{\text{génome}} = 3$ et $\delta_{\text{métabolique}} = 0$. Le temps de calcul nécessaire à l'obtention des résultats pour tous les organismes est de l'ordre de quelques minutes (moins de 5 secondes par organisme).

9.5.3.1 Résultats bruts

Les résultats globaux sont représentés sur la figure 9.9. Sur cette figure sont comparés le nombre de gènes de l'organisme, le nombre d'enzymes prédites (issues des annotations, HAMAP et de la prédiction) et le nombre d'enzymes prédites en opéron. Une enzyme est considérée comme étant en opéron si son gène fait partie d'un 2-uplet solution dont le nombre de gènes est égal ou supérieur à 2.

On remarque que le nombre d'enzymes prédites pour les organismes est à peu près linéaire avec le nombre de gènes total (coefficient de corrélation de 0.94). Pour la comparaison du nombre d'enzymes prédites en opérons avec le nombre de gènes et le nombre d'enzymes, la relation est moins marquée mais toujours significative (0.58 et 0.72 respectivement).

La figure 9.10 montre, avec les mnémoniques des organismes, la relation entre nombre de gènes et le nombre d'enzymes prédites en opéron.

On peut observer sur ce graphique que les génomes de ces organismes ne semblent pas avoir la même propension à être organisés en opérons. Parmi les organismes ayant le plus fort rapport nombre d'enzymes en opérons sur nombre de gènes, les quatre premiers sont les endosymbionts *Candidatus Blochmannia floridanus* et *Buchnera* (les trois). Les organismes avec le plus faible rapport sont *Wolbachia* (endosymbiont de *Drosophila melanogaster*), les deux archæbactéries *Aeropyrum pernix* et *Pyrococcus horikoshii* et le mycoplasme *Ureaplasma urealyticum*. L'ensemble des cyanobactéries semblent également avoir un rapport nombre d'enzymes en opérons sur nombre de gènes faible. Il est possible de comparer ces graphiques (en particulier le second) avec celui de la figure 6.11 du § 6.2.2.4. On remarquera par exemple, que pour *Escherichia coli*, le rapport $\left(\frac{\#\text{enzymes en opérons}}{\#\text{enzymes détectées}}\right)$ ne

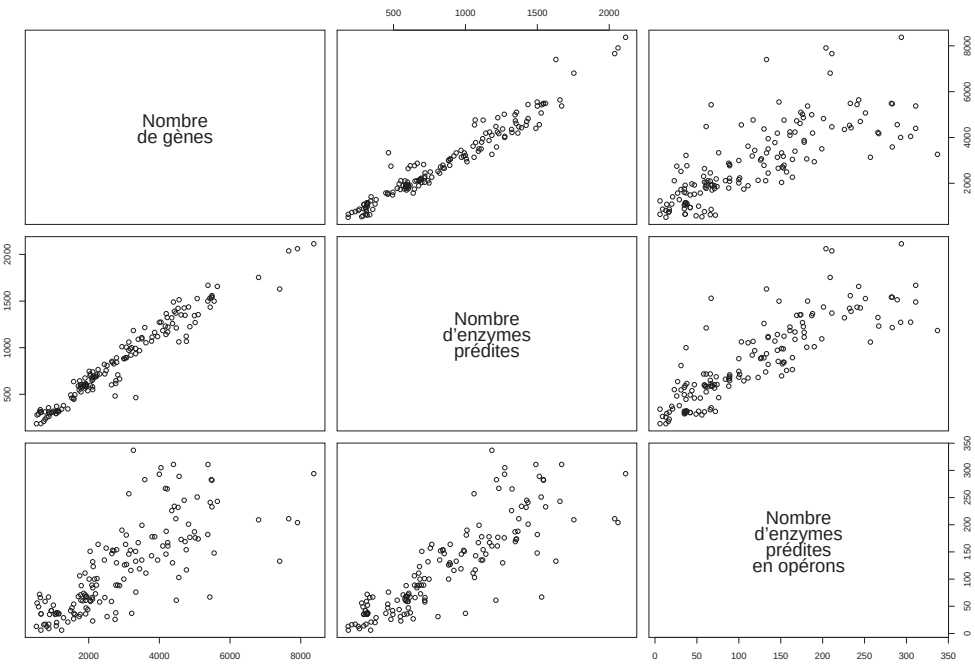


Fig. 9.9: Résultats globaux obtenus pour la prédiction d'opérons à partir de voies métaboliques - Chaque point correspond à un organisme. Ce graphique permet de mettre en évidence les corrélations fortes qui existent entre les 3 variables "Nombres de gènes" (le nombre total de gènes prédits pour l'organisme), "Nombres d'enzymes prédites" (le nombre de gènes associés à, au moins, un numéro EC) et "Nombres d'enzymes prédites en opérons" (la fraction des *enzymes prédites* qui sont associées à un opéron par la méthode)

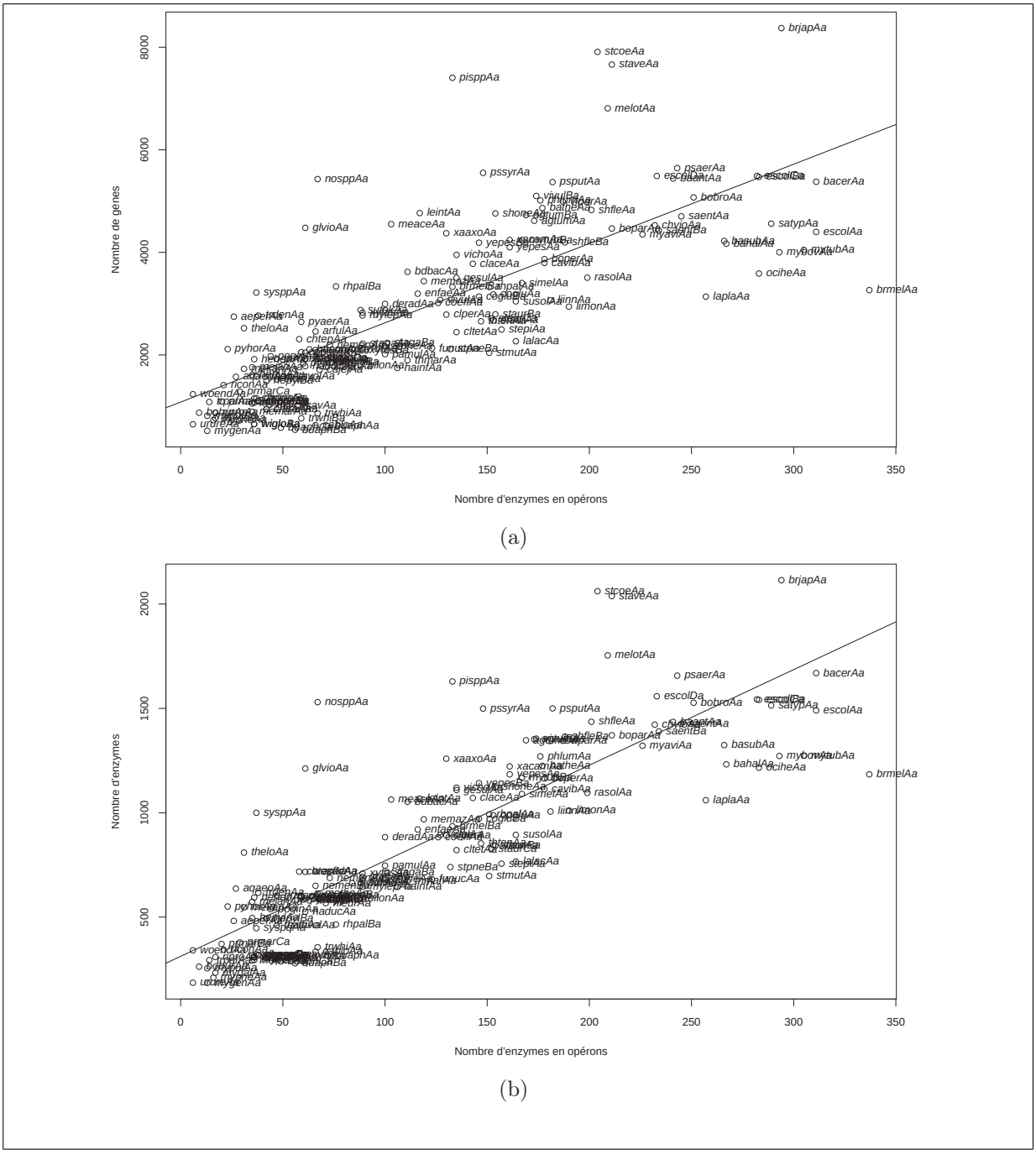


Fig. 9.10: Comparaison avec du nombre d'enzymes prédites en opérons avec (a) le nombre de gènes et (b) le nombre d'enzymes prédites pour chaque organisme

semble pas aussi important dans nos résultats que dans ceux présentés par [Zheng *et al.*, 2002].

Ces résultats doivent être tempérés par les annotations originelles des génomes et par la méthode de prédiction utilisée pour détecter les activités catalytiques des produits des gènes. En effet, pour le cas, par exemple d'*Aeropyrum pernix*, une étude attentive des annotations révèle, par exemple, des erreurs de prédictions des bornes des gènes. De plus, même si la banque ENZYME, sur laquelle repose la méthode de prédiction utilisée contient l'intégralité des enzymes connues, les enzymes ne sont pas toutes détectées, en particulier pour des organismes jusqu'ici moins étudiés que peuvent l'être les organismes modèles comme *Escherichia coli*. De plus, les conditions de formation des opérons sont strictes et on peut considérer que le nombre d'opérons trouvés ici est une borne inférieure du nombre de ce type d'opérons dans ces génomes.

9.5.3.2 Quelques voies métaboliques codées en opérons conservés chez les γ-protéobactéries

Le fait qu'une voie métabolique soit codée par un opéron dans un nombre important de génomes d'organismes différents permet d'attester de l'importance de cette voie et permet également de délimiter et de définir cette voie. Il peut donc être intéressant de rechercher ce type de voies.

En utilisant comme données un réseau réactionnel, plusieurs génomes et la restriction adéquate, le problème CCCMax permet de résoudre la requête suivante : **"Quelles sont les voies métaboliques qui sont codées sous la forme d'opérons dans tous ces organismes ?"**. Or, il peut arriver que dans un organisme, une voie métabolique ne soit pas présente ou que les gènes impliqués dans cette voie ne fasse pas partie du même opéron. Dans un tel cas, cette voie ne fera pas partie des résultats du problème CCCMax. Une façon de rendre le problème plus souple est d'introduire un quorum sur le nombre d'organismes concernés et d'exprimer la requête de la façon suivante : **"Quelles sont les voies métaboliques qui sont codées sous forme d'opérons dans au moins k parmi ces organismes ?"**. On pourrait modifier notre formulation initiale du problème CCCMax pour traiter ce cas. Néanmoins, nos premières expériences semblent montrer que le problème devient, en pratique, explosif.

Ces deux formulations sont illustrées sur la figure 9.11.

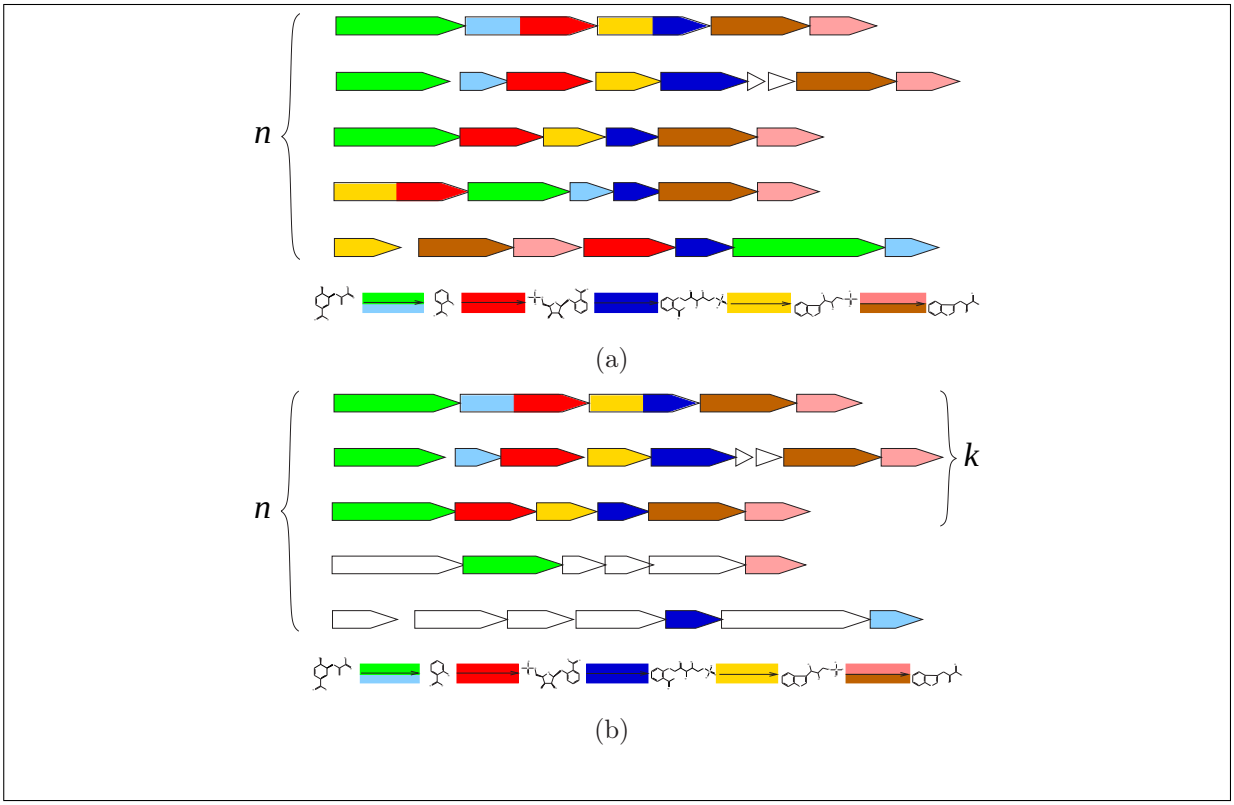


Fig. 9.11: Illustration des deux formulations du problème de la recherche d'opérons conservés dans plusieurs organismes - Dans la première formulation, on recherche, pour n génomes, quels sont les opérons présents dans tous ces génomes, dans la seconde formulation, on recherche les opérons présents dans au moins k sur les n génomes

Nous introduisons une formulation plus faible en considérant qu'une voie métabolique est caractérisée par un ensemble de numéros EC. Nous pouvons formuler la nouvelle question par : **"Quels sont les ensembles de numéros EC qui sont regroupés dans au moins k opérons différents ?"**. En pratique, cette question est voisine de la précédente.

Pour répondre à cette question, nous allons utiliser les prédictions d'opérons à partir du réseau métabolique pour chacun des génomes afin d'obtenir une collection d'opérons qui sont considérés comme autant d'ensembles de numéros EC. Ces ensembles de numéros EC subiront ensuite une procédure de filtrage afin d'obtenir un réponse à la question posée. La procédure complète pour trouver ces ensembles de numéros EC est illustrée sur la figure 9.12.

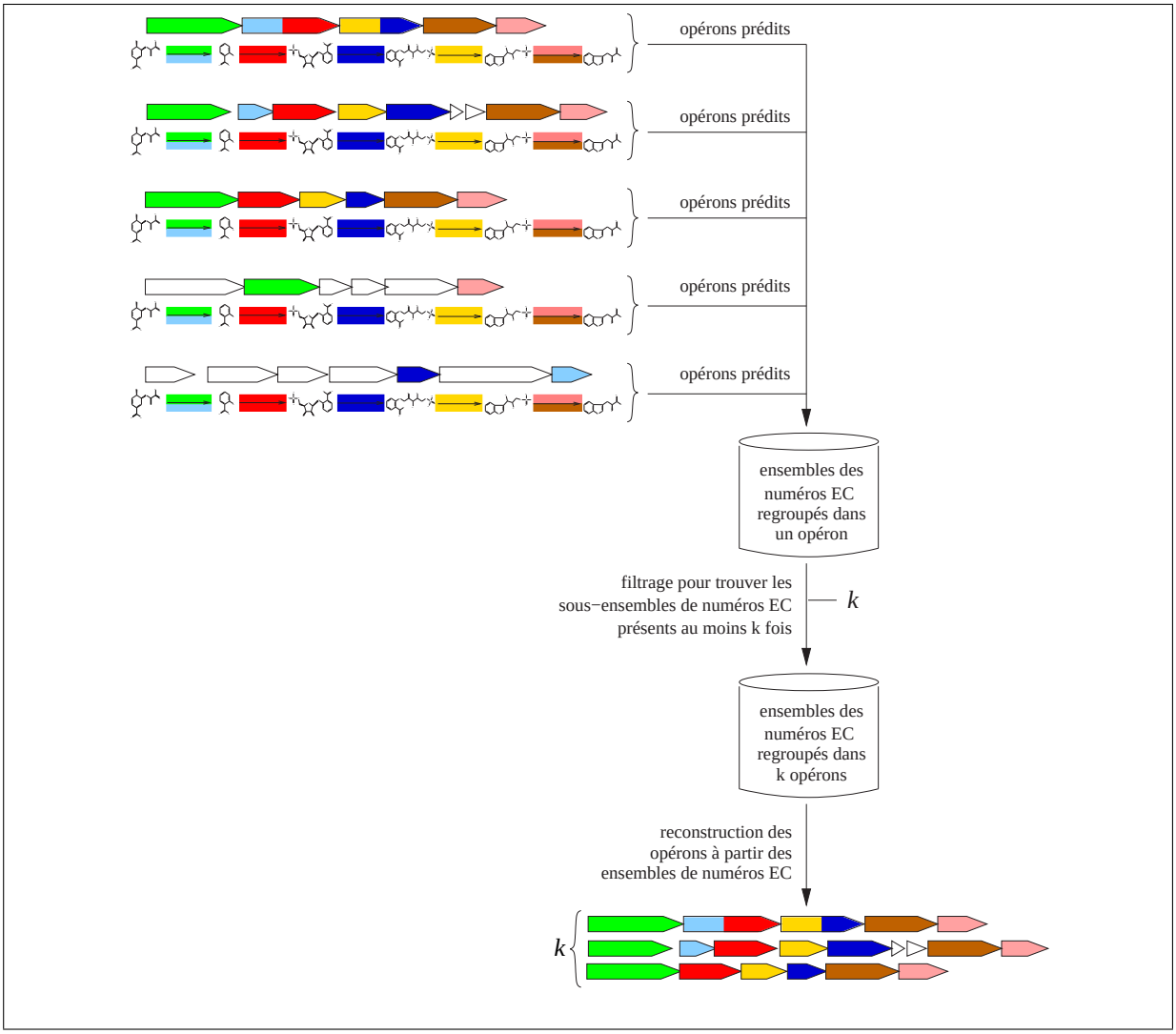


FIG. 9.12: Procédure pour l'obtention des opérons conservés dans k génomes - Dans un premier temps, l'intégralité des comparaisons des graphes des génomes contre le graphe des réactions est effectuée et tous les ensembles de numéros EC apparaissant dans un opéron sont regroupés. La deuxième étape consiste à trouver les ensembles de numéros EC qui apparaissent dans au moins k ensembles de numéros EC obtenus à l'étape précédente. A partir de ces ensembles de numéros EC, on reconstruit les opérons correspondants

La procédure de filtrage peut être traitée comme un problème d' "extraction d'itemsets fréquents" (une description de ce problème peut être trouvée, par exemple, dans [Goethals, 2003]).

L'application est ici limitée aux γ-protéobactéries. Les γ-protéobactéries sont des bactéries dont 33 génomes sont actuellement disponibles (dont 4 souches pour *Escherichia coli*).

Pour l'application, les ensembles d'EC groupés dans au moins $k = 20$ opérons parmi

tous les opérons prédits pour les 33 γ-protéobactéries disponibles ont été recherchés.

Dans la table 9.2, les groupes d'EC communs de taille supérieure à trois sont présentés avec la voie métabolique associée. On remarque que parmi ces voies-opérons conservées, trois concernent des acides aminés. Les deux plus grands groupes concernent la biosynthèse de la paroi cellulaire (peptidoglycane). Dans 18 organismes, l'ensemble des numéros EC de ces deux groupes sont regroupés dans un seul opéron. La biosynthèse de l'histidine a été séparée en deux car une des réactions (au milieu de la voie) n'est pas encore bien caractérisée et n'est associée à aucun numéro EC.

Liste d'EC	Voie métabolique
2.4.1.227 2.7.8.13 6.3.2.10 6.3.2.13 6.3.2.8 6.3.2.9	Biosynthèse du peptidoglycane
2.4.1.227 2.7.8.13 6.3.2.10 6.3.2.13 6.3.2.4	Biosynthèse du peptidoglycane
1.1.1.3 2.7.1.39 2.7.2.4 4.2.3.1	Biosynthèse de la thréonine
1.2.4.2 1.3.99.1 2.3.1.61 6.2.1.5	Cycle de Krebs (partiel)
1.1.1.23 2.6.1.9 3.1.3.15 4.2.1.19	Biosynthèse de l'histidine
1.1.1.11 1.1.1.57 1.1.1.58	Biosynthèse de l'histidine
1.1.1.193 2.5.1.9 3.5.4.26	Métabolisme de la riboflavine

Tab. 9.2: Numéros EC conservés en opérons dans les γ-protéobactéries

Les trois figures suivantes montrent, pour les voies de biosynthèse du peptidoglycane et du tryptophane, les gènes dont les activités catalytiques correspondent à celles conservées dans ces deux groupes. Ainsi, ces figures ne montrent pas uniquement l'opéron conservé mais également sa dispersion, et éventuellement sa disparition, dans les génomes des autres γ-protéobactéries.

La figure 9.13 montre, pour l'ensemble des γ-protéobactéries, présentées grâce à l'arbre des espèces donné par la taxonomie du NCBI [NCBI, 2004], les gènes possiblement impliqués dans la biosynthèse du peptidoglycane. On remarque que la structure de l'opéron est globalement très bien conservée. La figure 9.14 permet de mettre en évidence que, dans certains de ces organismes, l'organisation n'est toutefois pas complètement conservée, on y observe des déplacements de quelques gènes.

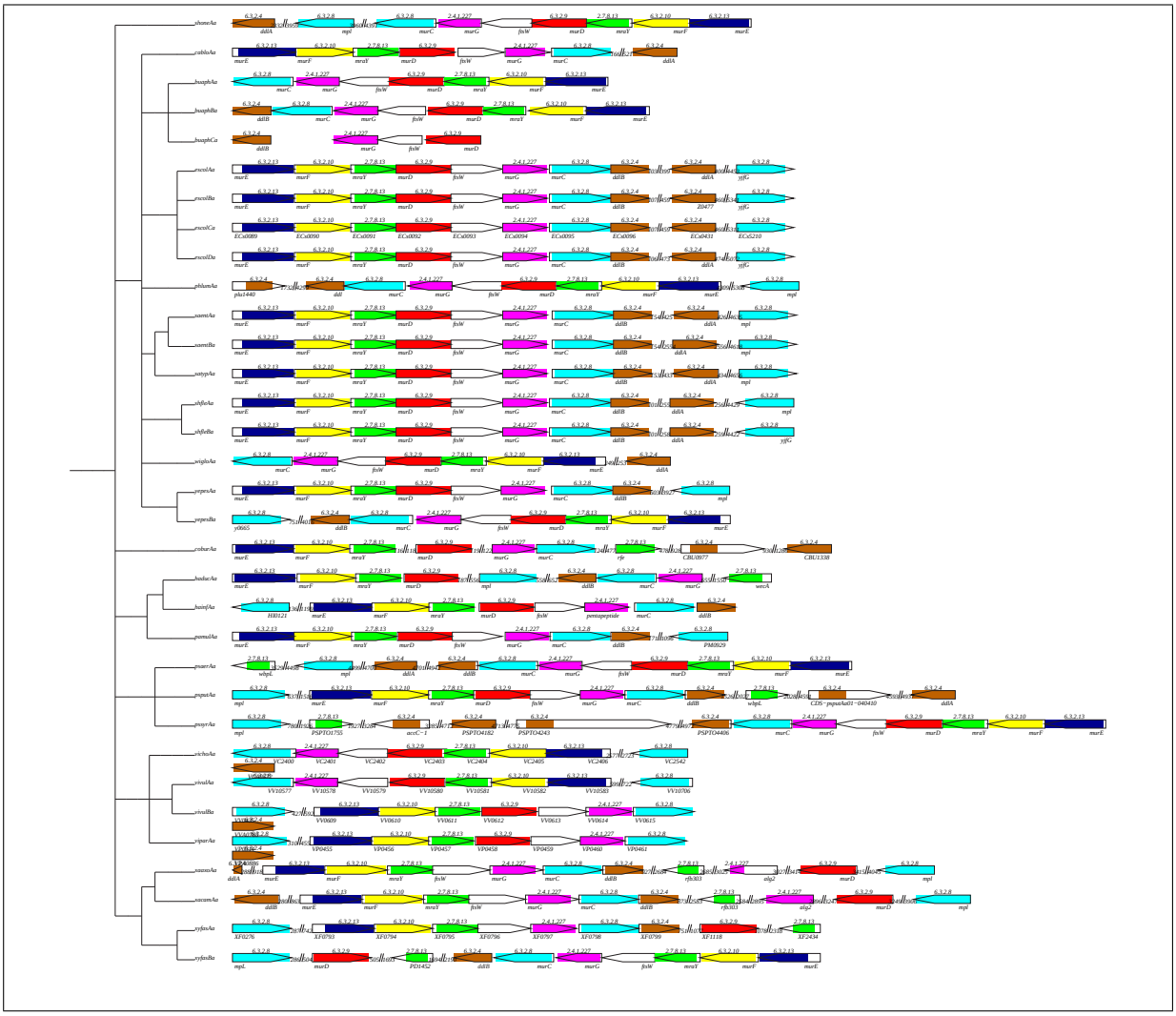


Fig. 9.13: Opéron conservé contenant les gènes impliqués dans la biosynthèse du peptidoglycane (numéros EC 2.4.1.227 2.7.8.13 6.3.2.4 6.3.2.8 6.3.2.9 6.3.2.10 6.3.2.13)

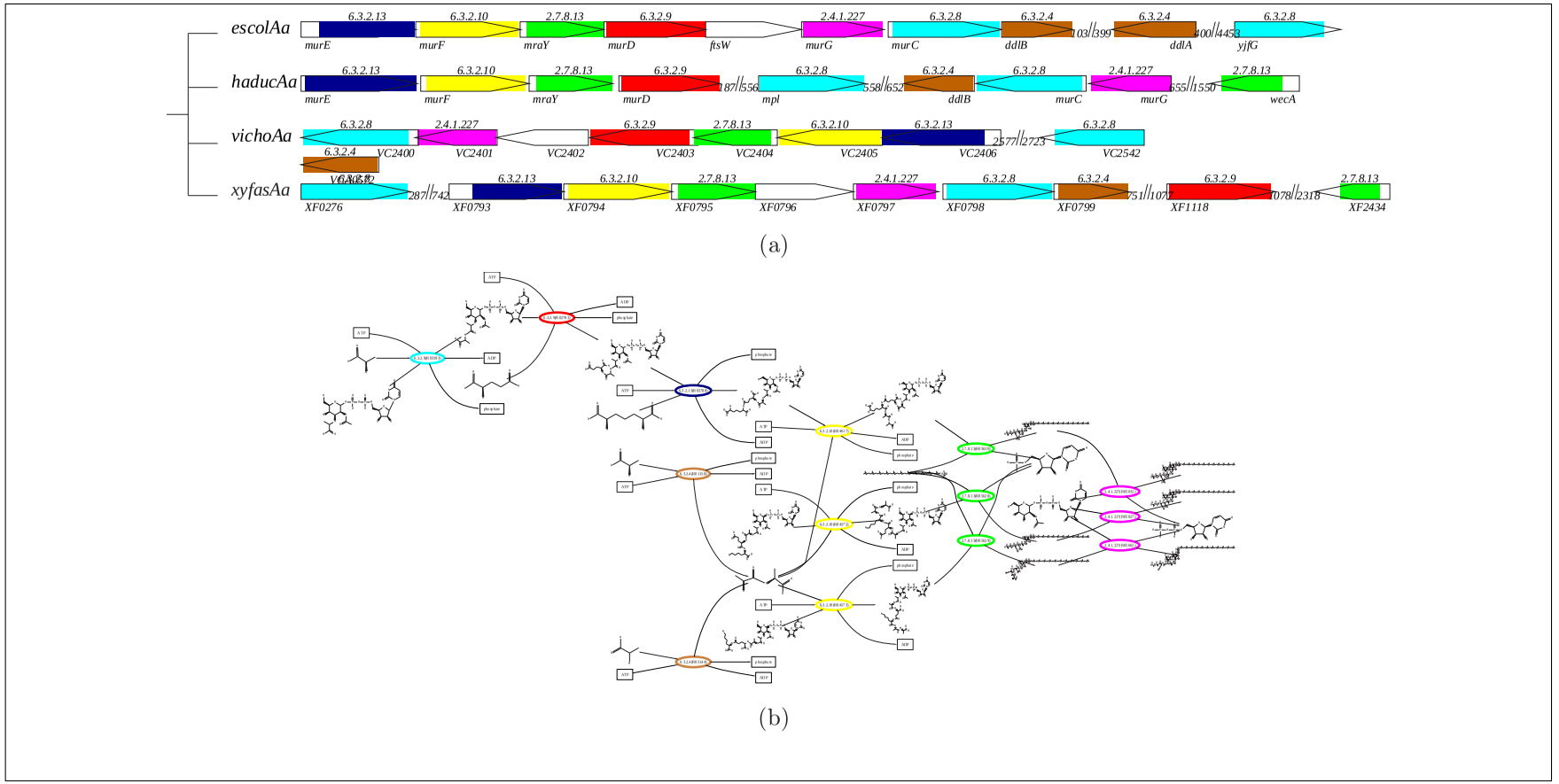


Fig. 9.14: Organisation des gènes associés à la biosynthèse du peptidoglycane (numéros EC 2.4.1.227 2.7.8.13 6.3.2.4 6.3.2.8 6.3.2.9 6.3.2.10 6.3.2.13) chez *Escherichia coli*, *Hæmophilus ducreyi*, *Vibrio cholerae* et *Xylella fastidiosa (9a5c)* et la correspondance des réactions dans la voie de biosynthèse du peptidoglycane - (a) l'opéron d'*Escherichia coli* contient l'intégralité des gènes associés à ces numéros EC, chez *Hæmophilus ducreyi* les gènes correspondants sont séparés en deux opérons, chez *Vibrio cholerae* un des gènes (EC 6.3.2.4) est sur le second chromosome tandis que chez *Xylella fastidiosa (9a5c)* c'est un autre gène (EC 6.3.2.9) qui ne fait pas partie de l'opéron, (b) voie de biosynthèse du peptidoglycane

Dans le cas de la biosynthèse du tryptophane (figure 9.15), on observe des phénomènes comme la fusion/dissociation de gènes (EC 4.1.1.48 et 5.3.1.24), l'insertion d'un ou plusieurs gènes, l'éclatement de l'opéron ou sa disparition complète. Cet opéron paraît moins stable que l'opéron codant pour la voie de biosynthèse du peptidoglycane, d'ailleurs, la liste des numéros EC correspondants ne fait pas partie du tableau 9.2 (tous ces numéros EC ne sont regroupés que dans 18 opérons pour les 33 γ-protéobactéries).

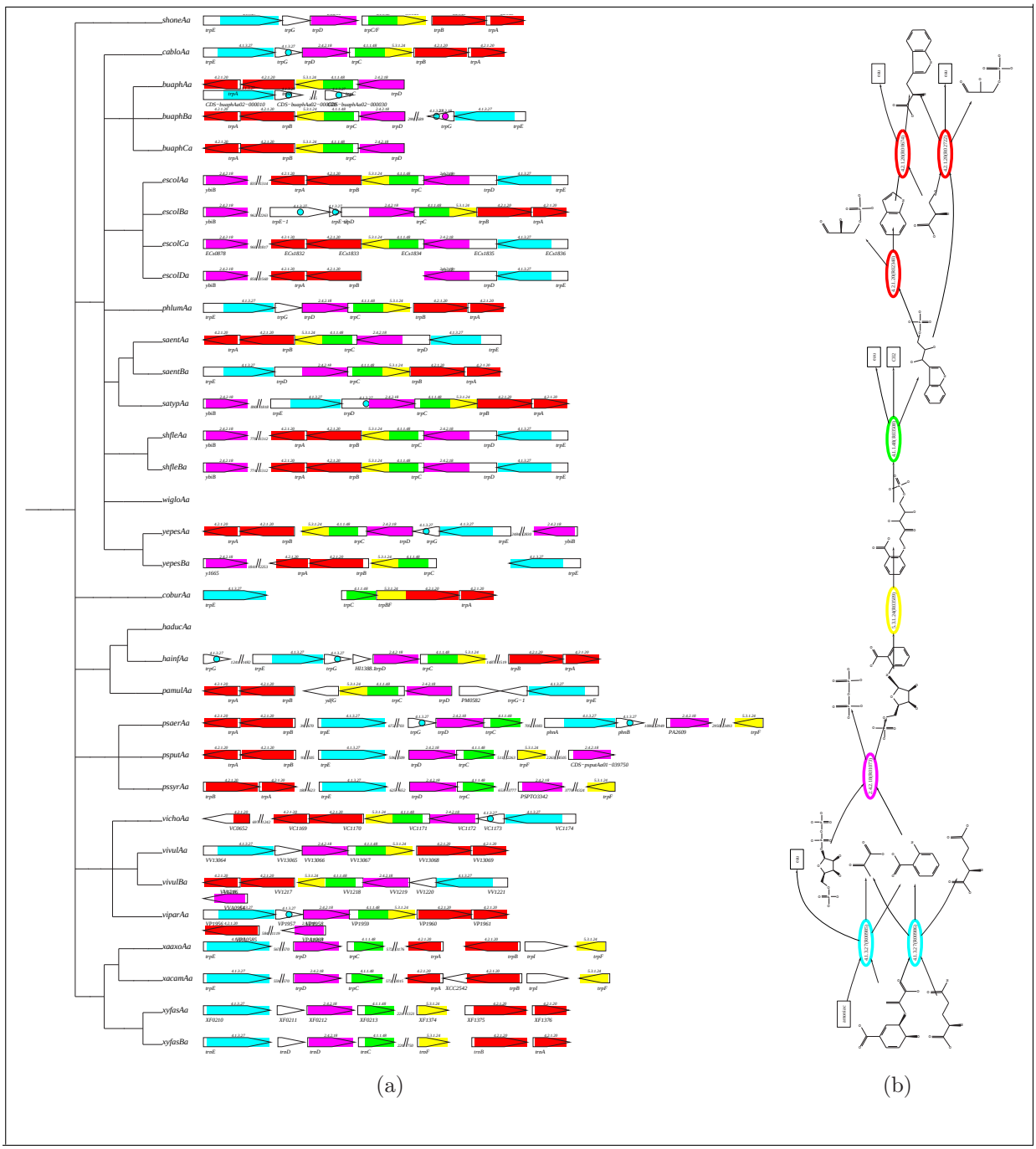


Fig. 9.15: Opéron conservé contenant les gènes impliqués dans la biosynthèse du tryptophane (numéros EC 2.4.2.18 4.1.1.48 4.1.3.27 4.2.1.20 5.3.1.24) - (a) on observe plusieurs disparitions de gènes à l'intérieur de plusieurs opérons prédits (qui sont peut-être des erreurs d'annotations) chez une souche d'*Escherichia coli* et *Coxiella burnetii (RSA 493)*, la disparition complète de la voie chez *Wigglesworthia glossinidia* (endosymbiont) et *Hæmophilus ducreyi (35000HP)* (pathogène) ainsi que la séparation des gènes constituant l'opéron dans de nombreuses bactéries, (b) voie de la biosynthèse du peptidoglycane

9.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés au problème de la comparaison des réseaux métaboliques et des génomes afin de relier les voies métaboliques aux structures en opérons. Ce problème avait déjà été abordé par [Ogata *et al.*, 2000; Zheng *et al.*, 2002]. Nous l'avons ici formalisé et généralisé sous la forme de la recherche de composantes connexes communes à plusieurs graphes.

Les applications effectuées illustrent l'intérêt de cette approche dans :

– la recherche d'opérons bactériens et archæbactériens

– la recherche de voies métaboliques conservées dans plusieurs espèces

Ceci peut s'étendre à d'autres réseaux ou relations. Notons en particulier, les réseaux d'interactions protéines-protéines, les réseaux de signalisation, ou encore les réseaux d'interactions géniques (pour d'autres exemples voir [Yanai and DeLisi, 2002]).

Par exemple, la comparaison des réseaux d'interactions protéines-protéines avec les réseaux métaboliques pourrait permettre de répondre à des questions telles que : **"Parmi toutes les protéines connues expérimentalement pour interagir, qu'elles sont celles qui interviennent dans des réactions métaboliques connexes ?"**. Dans ce cas, la relation permettant de connecter les protéines aux réactions est à nouveau donnée par le numéro EC.

La limite fondamentale de cette approche réside dans la nécessité de disposer d'une relation entre les nœuds des différents graphes.

Conclusion

Dans cette thèse, nous avons cherché à avoir la démarche la plus rigoureuse possible quant au traitement des questions biologiques traitées. Cette démarche consiste dans un premier temps à poser formellement le problème en identifiant bien les hypothèses qu'implique, au niveau biologique, cette formalisation. Dans un second temps, la résolution informatique du problème permet de valider ou d'invalider ces hypothèses.

Bilan

Au cours de cette thèse, deux problèmes différents ont été abordés :

- la reconstruction de chemins réactionnels sur la base d'un ensemble de réactions
- l'identification des relations entre l'organisation de réseaux métaboliques et l'organisation physique des gènes sur le chromosome

En ce qui concerne le premier problème, et qui constitue le corps de ce travail de thèse, une nouvelle formalisation a été proposée pour la reconstruction de voie métabolique. Cette formalisation repose sur la définition suivante : **Une voie métabolique entre deux composés est une succession de réactions qui transfère le plus grand nombre d'atomes du composé initial vers le composé final**. Le problème de la recherche de voies métaboliques peut alors être formalisé comme celui de la recherche d'une composition d'injections partielles, entre deux ensembles donnés, dont il faut maximiser la taille de l'image.

Un algorithme résolvant exactement ce problème a été proposé et a été implémenté. Les premiers résultats obtenus sur la reconstruction de voies métaboliques connues sont encourageants, tant en termes biologiques qu'en termes informatiques (temps de calcul). Néanmoins, cette approche nécessite encore d'être validée par les biologiques eux-mêmes. Dans ce but, une utilisation interactive du programme devient nécessaire. L'intégration du programme dans un environnement interactif devrait permettre, à terme, cette validation. Un travail d'intégration dans l'environnement d'expertise de données génomiques et post-génomiques GEB (*GenoExpertBacteria*), développé dans l'équipe par Anne Morgat, est actuellement en cours.

Pour ce qui concerne la seconde partie, nous avons tenté de formaliser un problème pour lequel aucune définition vraiment claire n'existait. Nous avons montré que la question pouvait se ramener, moyennant une transformation des graphes initiaux, au problème de la recherche de composantes connexes communes.

L'application à l'étude des relations entre l'organisation de réseaux métaboliques et celle des gènes sur le chromosome permet de se faire une idée plus précise de la propension des génomes bactériens à s'organiser en opérons sous l'effet de pressions fonctionnelles ou,

réciproquement, de rechercher les voies métaboliques possibles à partir d'informations de colocalisation chromosomique.

Perspectives

Nous présentons ici quatre extensions envisageables sur ce travail.

Valider les chemins réactionnels

La première extension concerne une restriction importante notée au chapitre 8 et dans la conclusion de l'article donnée en annexe A. Notre méthode de reconstruction n'impose que de faibles contraintes (pas de condition stoechiométrique ou thermodynamique) sur les solutions. Le critère employé (transfert maximal d'atomes) peut être trop simple et conduire à des solutions thermodynamiquement irréalistes. Il pourrait donc être important de fournir des moyens de valider ces chemins réactionnels.

Lorque ceci est possible, on pourrait intégrer des informations relatives à la cinétique des réactions afin d'avoir une idée sur la vraisemblance thermodynamique des chemins réactionnels proposés. Dans un premier temps, l'intégration des informations concernant la réversibilité des réactions permettrait probablement d'éviter de proposer un certain nombre de chemins qui, de fait, ne sont pas pertinents.

Un autre moyen, pour augmenter la qualité des prédictions, serait d'y intégrer des résultats expérimentaux. Par exemple, si l'on a une connaissance partielle de la voie recherchée (présence ou absence d'une réaction particulière), il est tout à fait possible d'imposer aux chemins réactionnels de passer par ces réactions. Un type d'expérience facile à intégrer dans notre formalisation consiste à marquer, par radioactivité, certains atomes de composés présents dans le milieu de culture et à observer ensuite la distribution d'atomes marqués dans les composés présents dans la cellule. En effet, si l'on sait quels atomes du composé initial se retrouvent dans le composé final, cette information permet de réduire grandement le nombre de chemins réactionnels en ne recherchant que les successions de réactions qui transfèrent ces atomes.

Prise en compte des bases de réactions incomplètes

L'approche que nous avons proposée repose sur la connaissance d'une base de réactions sinon exhaustive, tout du moins complète vis à vis du problème traité. Malheureusement dans la réalité, les bases de données disponibles (KEGG) sont rarement exhaustives ou totalement cohérentes. La question se pose donc de prendre en compte les bases de réactions incomplètes.

Une première direction consiste à enrichir la base de réactions inférées.

Pour inférer ces réactions, on pourrait procéder en plusieurs étapes. La première étape serait de constituer des groupes de réactions similaires (par exemple phosphorylation) mais qui s'appliquent à des composés chimiques un peu différents. Cette première étape pourrait être réalisée en classant les réactions d'abord par numéros EC similaires puis par groupes de composés similaires. La similarité entre composés étant obtenue en comparant les structures moléculaires des composés (de la même façon que nous l'avons effectué au § 8.3).

La seconde étape consisterait à établir des règles décrivant chaque groupe de réactions. Ces règles pourrait avoir la forme de patrons (patterns) structuraux vérifiés par tous les composés du groupe (par exemple présence d'un ribose).

Enfin, l'inférence d'une nouvelle réaction reposerait sur l'application de ces règles à des composés satisfaisant les règles mais pour lesquels la réaction associée n'est pas connue.

La figure 9.16 illustre la succession de ces trois étapes.

Cette approche pourrait se généraliser non pas à une seule réaction mais à une succession de réactions permettant d'aller d'une molécule à une autre comme l'illustre la figure 9.17. La première étape consisterait alors à identifier un couple de composés proches du couple initial (en se basant sur la similarité de leur graphe moléculaire). Pour ce couple, on rechercherait ensuite un chemin réactionnel. Enfin, ce chemin réactionnel devra être "adapté" au couple initial de composés.

Etendre la notion de transfert entre composés

Récemment, [Nobeli *et al.*, 2003] ont montré que les composés impliqués dans les mêmes voies métaboliques avaient tendance à se ressembler structuralement. Cette observation va dans le sens de l'hypothèse sur laquelle repose notre formulation d'un chemin réactionnel, à savoir qu'un chemin conserve au mieux la structure de la molécule de départ dans la molécule d'arrivée.

Dans [Nobeli *et al.*, 2003], plusieurs mesures de similarité entre composés sont utilisées. L'une d'elles est basée sur la présence, dans la structure des molécules, de sous-structures caractéristiques de fonctions chimiques. Les molécules sont ainsi décrites à l'aide d'une librairie de 57 motifs structuraux représentée sur la figure 9.18.

La similarité entre deux molécules utilise alors une description booléenne associée à la présence ou l'absence de chacun de ces 57 motifs structuraux. La similarité entre deux molécules est simplement définie comme le nombre de motifs structuraux communs.

Sur la base de cette observation, une extension possible de notre approche serait, non plus de rechercher les chemins réactionnels qui transfèrent un nombre maximum d'atomes, mais ceux qui conservent un nombre maximum de motifs structuraux (représentant des

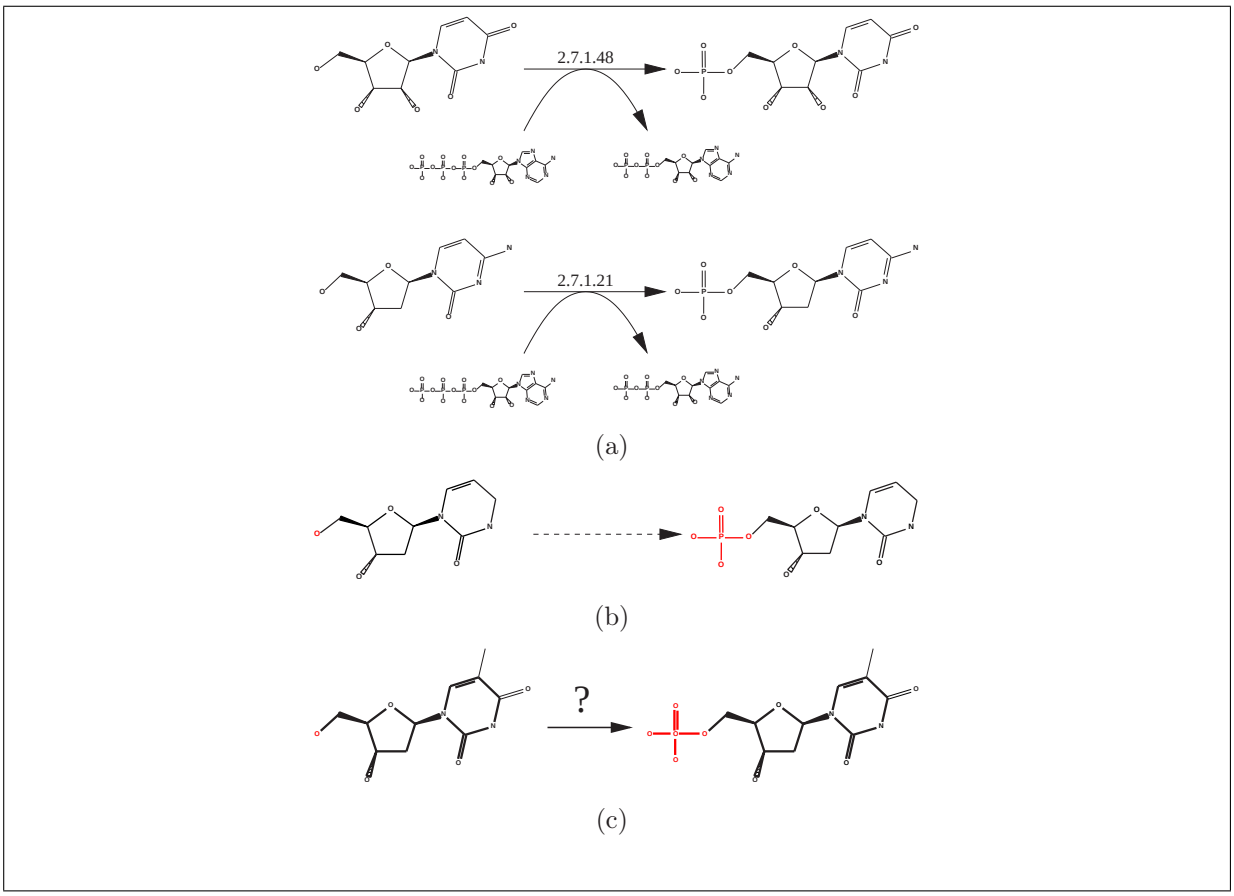


Fig. 9.16: Inférence d'une nouvelle réaction - La première sous-figure (a) montre deux réactions de phosphorylation pour lesquelles on décide d'associer le pattern structural (b). Le couple de composés montré en (c) satisfait le patron structural, on peut donc inférer que la réaction est possible

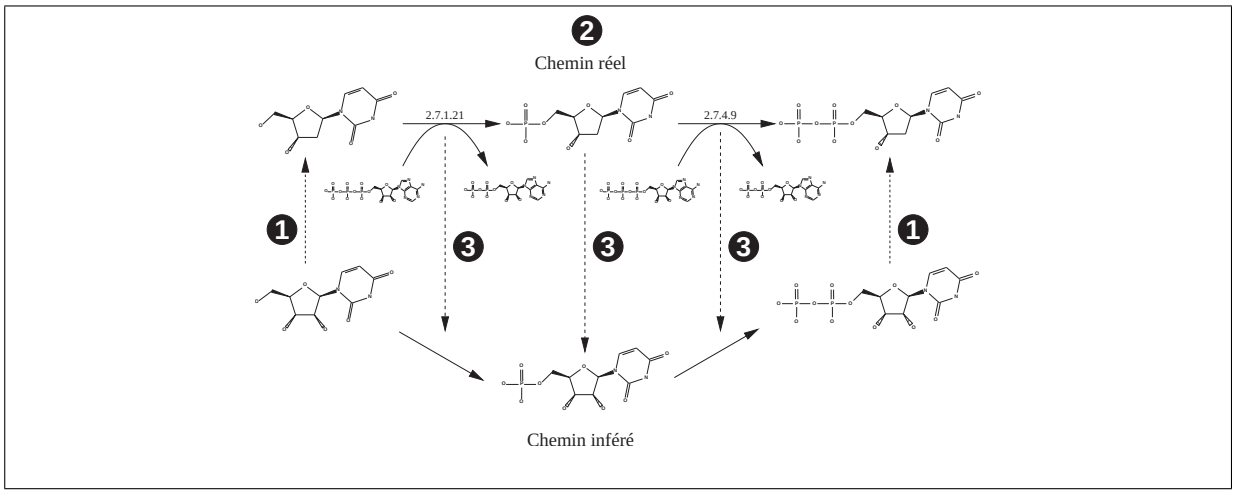


Fig. 9.17: Généralisation de l'inférence des réactions manquantes à des chemins métaboliques complets - On peut distinguer trois étapes dans ce processus. La première étape (1) consiste à chercher un couple de composés *similaires* au couple de composés initiaux. Pour ce nouveau couple de composés, on recherche ensuite un chemin métabolique (2). Enfin, un nouveau chemin métabolique est déduit de ce chemin (3) en inférant à la fois les réactions et les composés manquants

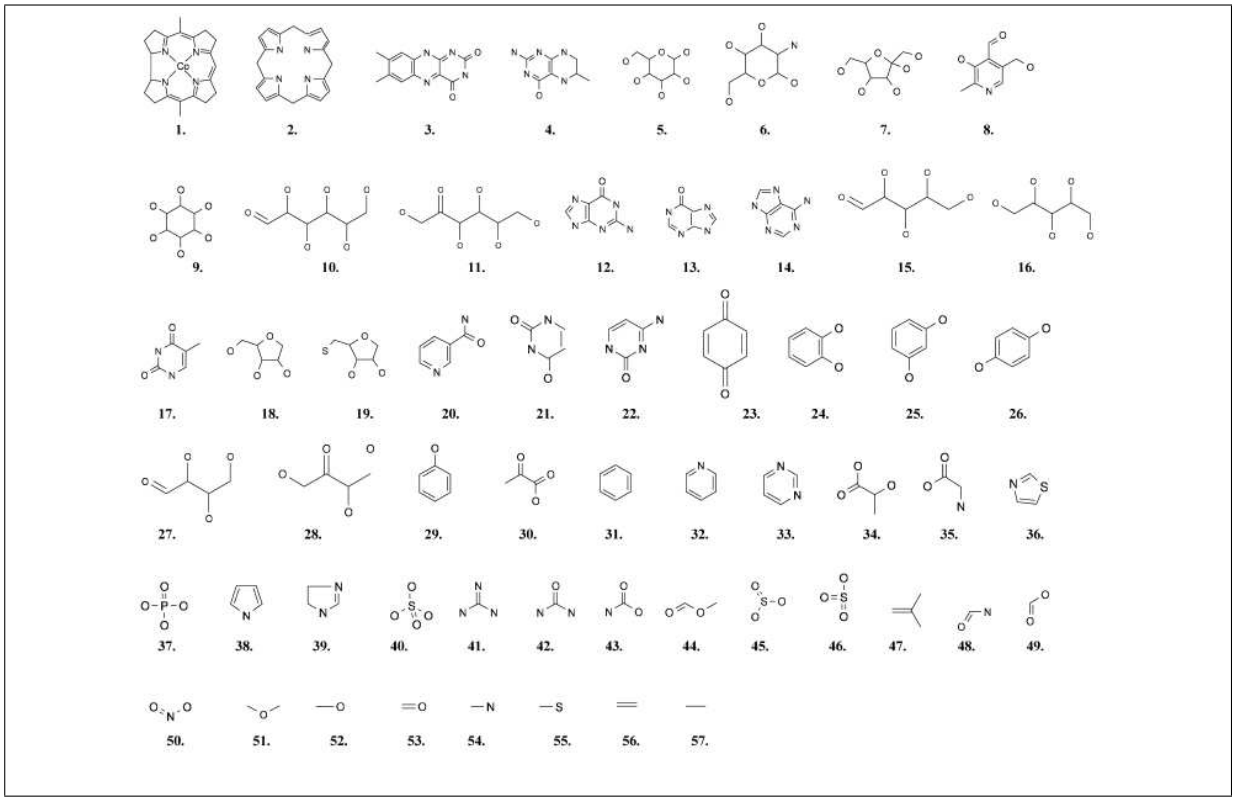


Fig. 9.18: La librairie de motifs structuraux FRAG57 utilisée dans [Nobeli *et al.*, 2003]

fonctions) entre deux molécules. Cela apportera peut-être une meilleure robustesse des prédictions, le nombre d'atomes transférés n'étant pas forcément le meilleur critère d'évaluation.

Couplage de la recherche de chemins réactionnels avec la recherche d'opérons

Enfin, il serait tout à fait possible d'utiliser l'approche décrite au § 8.3 comme un outil pour restreindre les réactions utilisées dans la recherche des chemins réactionnels.

En effet, lors de la reconstruction d'une voie métabolique partagée entre plusieurs organismes, on peut émettre l'hypothèse que les enzymes sont également conservées entre ces organismes et que les gènes correspondants sont organisés de la même façon. On pourrait donc, dans un premier temps, rechercher les groupes d'enzymes qui sont co-localisées sur tous les chromosomes de ces organismes et utiliser ces groupes pour restreindre l'ensemble des réactions pris en compte dans la phase de reconstruction.

L'hypothèse forte est ici que l'organisation des gènes est conservée dans tous les organismes. Plutôt que de faire intervenir cette hypothèse avant la phase de reconstruction, il serait probablement plus réaliste de la faire intervenir à posteriori, c'est-à-dire de l'utiliser en filtre sur l'automate produit. En effet, dans l'automate, les transitions correspondent à des réactions qui relient des composés. Le graphe des arêtes associé à l'automate est donc un graphe réactionnel. On pourrait comparer ce graphe à l'organisation chromosomique des organismes étudiés. Ainsi, si dans l'automate un chemin est pris en charge par des enzymes codées en opérons, ce chemin devient un bon candidat et doit être étudié avec une attention toute particulière.

Annexe A

Article paru dans la revue Bioinformatics

BIOINFORMATICS

Vol. 19 Suppl. 2 2003, pages ii26–ii34
DOI: 10.1093/bioinformatics/btg1055

Ab initio reconstruction of metabolic pathways

Frédéric Boyer[1,2] and Alain Viari[2,*]

[1]LSR-IMAG, 681, rue de la Passerelle, BP. 72, 38402 Saint Martin d'Hères Cedex, France and [2]INRIA Rhône-Alpes, 655, avenue de l'Europe, Montbonnot, 38334 Saint Ismier Cedex, France

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT

We propose a new formulation for the problem of *ab initio* metabolic pathway reconstruction. Given a set of biochemical reactions together with their substrates and products, we consider the reactions as transfers of atoms between the chemical compounds and we look for successions of reactions transferring a maximal (or preset) number of atoms between a given source and sink compound. We state this problem as the one of finding a composition of partial injections that maximizes the image size. First, we study the theoretical complexity of this problem, state some related problems and then give a practical algorithm to solve them. Finally, we present two applications of this approach to the reconstruction of the tryptophan biosynthesis pathway and to the glycolysis.

Contact: alain.viari@inrialpes.fr

INTRODUCTION

Reconstructing metabolic pathways of fully sequenced organisms is becoming a task of major importance and several approaches have already been proposed in order to help biologists in identifying and analyzing the metabolic pathways of a newly sequenced organism. A first line of approach relies on a database of already characterized metabolic pathways. Then, for each known pathway (or part of a pathway) of this database, one has to find if it *occurs* in the organism under study. To occur means, for instance, that the genes encoding for the catalysts (i.e. the enzymes) are annotated in the genome under study (Gaasterland *et al.*, 1995; Kanehisa, 1999; Karp *et al.*, 1999; Overbeek *et al.*, 1999). This approach yields good results (Paley *et al.*, 2002) but is, of course, unable to predict unknown or alternative metabolic pathways. Moreover, it strongly relies on the correct annotation of the genome under study. Another, more exploratory approach, is the *ab initio* metabolic pathway reconstruction problem which consists in finding putative metabolic pathways connecting a set of given compounds without any other knowledge than a set of reactions (and the chemical compounds they involve) (Arita, 2000; Küffner *et al.*, 2000; Mavrovouniotis, 1993; Schilling *et al.*, 2000; Schuster *et al.*, 2000). By contrast, this approach allows to predict new, possibly unrealistic, pathways that should be further assessed.

The most widespread approach in *ab initio* metabolic pathway reconstruction considers compounds as resources and reactions as rules combining compounds (Küffner *et al.*, 2000; Mavrovouniotis, 1993; Schilling *et al.*, 2000; Schuster *et al.*, 2000). Briefly, the idea is, to find sets of reactions satisfying some *flux balance* conditions. For instance, one should produce a specified *sink* compound from a *source* compound, keeping a balanced overall production and consumption of all intermediate compounds. Although they give rise to very elegant and compact formulations, these approaches may encounter practical difficulties with some specific, usually highly connected, compounds such as cofactors, coenzymes, water also called *ubiquitous substrates*. Considering them as intermediate compounds tends to make the solutions grow (i.e. to aggregate several pathways). One should therefore either ignore them or consider them as always available for production or degradation. The main difficulty is then that the solution greatly depends upon the arbitrary definition of this set of ubiquitous substrates. It should also be pointed out that, in this approach, chemical compounds are simply considered to as labels (i.e. their atomic structure is ignored).

In this paper, we propose an alternative approach to the *ab initio* metabolic pathway reconstruction problem. The main idea is to consider chemical compounds as sets of individual atoms and reactions as transfers (partial injections) of atoms between compounds (a similar point of view was already introduced in Arita (2000)). Given a source and sink compound, the reconstruction problem consists in finding all the successions of reactions that result in a minimum number of transferred atoms from the source to the sink (this threshold constitutes the only parameter of the approach). By design, this approach does not take into account any *flux balance* condition and therefore does not require the arbitrary definition

*To whom correspondence should be addressed.

of ubiquitous substrates. On the other hand, it may produce more thermodynamically unrealistic solutions (as we shall see in the last section, it does not produces so unrealistic solutions however). Therefore this approach should actually be considered as complementary to the previous one.

The remainder of this paper is divided in three parts. We shall present a general sketch of our approach and reduce it to a problem of composing partial injections. Then we shall study the theoretical complexity of this and related problems. We also present a practical algorithm to compute an automaton accepting all valid compositions. Finally, the last section presents two practical applications of this approach.

SKETCH OF THE APPROACH

Our approach can be split into two successive problems:

1. define a unique mapping between atoms on each side of a reaction;
2. compute, on the basis of these mappings for all reactions, all paths ensuring a minimum transfer of atoms between given source and sink compounds.

The first problem is classical in the area of computational chemistry and we shall just recall it here. This problem can be expressed as a MAXIMUM COMMON SUBGRAPH problem (Crescenzi *et al.*, 1998, GT46) that is the problem of finding an isomorphism between two graphs by deleting the minimum number of edges. In our case, the two graphs (called molecular graphs) correspond to each side of the reaction. Figure 1 shows an example of such graphs with the reaction: 2-acetolactate + $CO_2 \leftrightharpoons$ 2 pyruvate. In a molecular graph, vertices represent atoms and edges represent bonds, both edges and vertices are labeled, with the atom type for the vertices and with the bond type for the edges. The molecular graph representing the left (resp. right) side of a reaction is the union of the molecular graphs of the compounds involved as substrates (resp. products) of the reaction (see Fig. 1a and b), these two molecular graphs have therefore the same number of vertices. A solution of the MAXIMUM COMMON SUBGRAPH problem is an isomorphism, that is a one-to-one correspondence between vertices of each molecular graph (see Fig. 1c), that minimizes the total number of broken bonds during the reaction. Coming back to the individual compounds, a solution therefore defines a partial injection (possibly empty) between all possible (substrate, product) couples of the reaction (see Fig. 1d). These partial injections have to be calculated once for all on a database of reactions. They form the input of the second problem that is, to compose these injections in order to obtain all paths between a given source and sink compound (see Fig. 2) that result in a sufficient transfer of atoms.

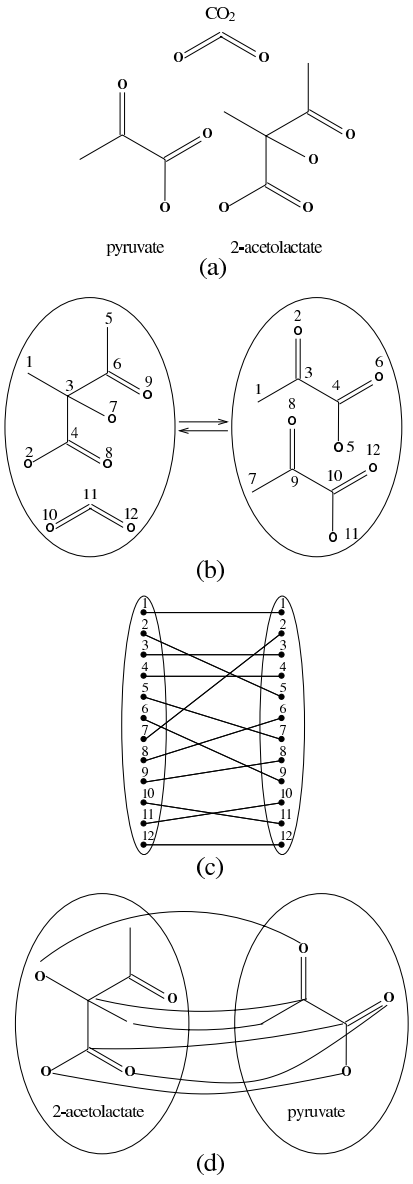


Fig. 1. (a) The molecular graphs of CO_2, pyruvate and 2-acetolactate, (b) The graphs associated to the two sides of the reaction 2-acetolactate + $CO_2 \leftrightharpoons$ 2 pyruvate, (c) A one-to-one correspondence between the nodes of the two graphs representing a possible transfer of atoms between these compounds and (d) The transfer of atoms (partial injection) between 2-acetolactate and pyruvate deduced from (c).

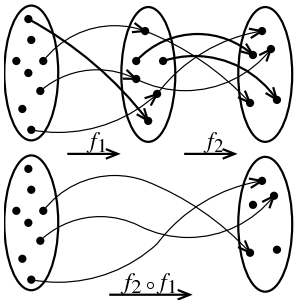


Fig. 2. The partial injection resulting from the composition of two partial injections.

Although the first problem (MAXIMUM COMMON SUBGRAPH) is known to be NP-hard (Crescenzi *et al.*, 1998), several efficient heuristics have already been designed for solving it, especially in the context of molecular graphs (Arita, 2000). In addition, recent results have shown that, for special classes of reactions, this problem is solvable in polynomial time (Akutsu, 2003). In the next section, we shall focus on the second problem, that is, to compose the partial injections.

THE MAXIMAL PARTIAL INJECTIONS COMPOSITION AND RELATED PROBLEMS

We start by stating the problem of finding a composition of partial injections of maximal size between two sets given a collection of partial injections. We then focus on the complexity of this problem and state more practical problems derived from the basic one. We finally give an algorithm to solve these problems.

The maximal partial injections composition problem

Given a partial injection $I : X \to Y$, we define its *size* as $Size(I) = |I(X)|$, that is the number of $x_i \in X$ that have an image in Y

PROBLEM 1. PARTIAL INJECTIONS COMPOSITION *(PIC)*
INSTANCE*: A collection of n distinct sets* $\mathcal{X} = \{X_1, \ldots, X_n\}$*, a collection of m partial injections (defined on sets in* $\mathcal{X}$*)* $\mathcal{I} = \{I_1, \ldots, I_m\}$
QUESTION*: Is there a composition of partial injections* $I_{comp} \in \mathcal{I}^*$ *from* X_1 *to* X_n *such that* $Size(I_{comp}) = min(|X_1|, |X_n|)$*?*

PROPOSITION 1. *The PIC problem is PSPACE-Complete*

To prove PSPACE-completeness of the PIC problem we reduce instances of a special case of the FINITE STATE AUTOMATA INTERSECTION problem to instances of PIC. The FINITE STATE AUTOMATA INTERSECTION problem has been proven PSPACE-complete in general (Kozen, 1977). For the reduction, we use a result from Birget *et al.* (2000) which states that the problem remains PSPACE-complete for a special type of automata.

PROOF. PIC is in NPSPACE since we can guess a sequence $s \in \mathcal{I}^*$ of partial injections, apply successively the injections on the successive results beginning with the set X_1, and refuse if we cannot compose two successive injections (because of definition domains disagreement) or if we finish with a subset of X_n of size less than $min(|X_1|, |X_n|)$, and accept otherwise. As a result of Savitch's theorem (Savitch, 1970), PIC is therefore in PSPACE.

DEFINITION 1. *Injective automaton*
Let $\mathcal{A} = (Q, \Sigma, \delta, i, F)$ *be a finite deterministic state automaton defined by its finite set of states Q, a finite set of symbols* Σ*, a transition function* $\delta : Q \times \Sigma \to Q$*, its initial state i and its set of final states* $F \subseteq Q$*.*
We say that $\mathcal{A}$ *is injective iff every symbol* $\sigma \in \Sigma$ *induces a partial one-to-one function on Q and* $|F| = 1$*.*

PROBLEM 2. INJECTIVE FINITE STATE AUTOMATA INTERSECTION
INSTANCE*: A sequence* $\mathcal{A}_1, \ldots, \mathcal{A}_n$ *of injective deterministic finite state automata having the same input alphabet* Σ
QUESTION*: Is there a string* $x \in \Sigma^*$ *accepted by each of the* $\mathcal{A}_i$*,* $1 \leq i \leq n$*?*

THEOREM 1. *The* INJECTIVE FINITE STATE AUTOMATA INTERSECTION *problem is PSPACE-Complete (Birget* et al.*, 2000)*

We now reduce the PIC problem from the INJECTIVE FINITE STATE AUTOMATA INTERSECTION problem. Let $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ be a sequence of n injective automata having the same input alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_{|\Sigma|}\}$, with $\mathcal{A}_i = \left(Q_{\mathcal{A}_i}, \Sigma, \delta_{\mathcal{A}_i}, q_{\mathcal{A}_i}^{init}, \left\{q_{\mathcal{A}_i}^{final}\right\}\right)$, $Q_{\mathcal{A}_i} = \left\{q_{\mathcal{A}_i}^1, \ldots, q_{\mathcal{A}_i}^{|Q_{\mathcal{A}_i}|}\right\}$, $q_{\mathcal{A}_i}^{init} \in Q_{\mathcal{A}_i}$ and $q_{\mathcal{A}_i}^{final} \in Q_{\mathcal{A}_i}$
One must define 3 sets X_1, X_2 and X_3:

- $X_1 = \{x_1^1, \ldots, x_1^n\}$
- $X_2 = \bigcup_{i=1}^{n} Q_{\mathcal{A}_i}$
- $X_3 = \{x_3^1, \ldots, x_3^n\}$

and $2 + |\Sigma|$ partial injections:

- I_{init} from X_1 to X_2: $I_{init}\left(x_1^i\right) = q_{\mathcal{A}_i}^{init}, 1 \leq i \leq n$
- I_{final} from X_2 to X_3: $I_{final}\left(q_{\mathcal{A}_i}^{final}\right) = x_3^i, 1 \leq i \leq n$
- I_{σ_i} $(1 \leq i \leq |\Sigma|)$ from X_2 to X_2: $\forall k, 1 \leq k \leq |\mathcal{A}|$ and $\forall l, 1 \leq l \leq |Q_{\mathcal{A}_k}|$, $I_{\sigma_i}\left(q_{\mathcal{A}_k}^l\right)$ is defined if $\delta_{\mathcal{A}_k}\left(q_{\mathcal{A}_k}^l, \sigma_i\right)$ is defined and $I_{\sigma_i}\left(q_{\mathcal{A}_k}^l\right) = \delta_{\mathcal{A}_k}\left(q_{\mathcal{A}_k}^l, \sigma_i\right)$

Let's suppose that the intersection of the n injective automata is nonempty and the word $w = \sigma_i \ldots \sigma_j$ belongs to this intersection, then the composition $I_{comp} = I_{final} \circ I_{\sigma_j} \circ \ldots \circ I_{\sigma_i} \circ I_{init}$ is a solution of the constructed instance of PIC, the opposite is also true. □

NOTE 1. *Under its most general form, a solution of the PIC problem may involve the same set X_i several times. If we restrict the problem to the case where all sets should be used at most once, the PIC problem is NP-complete (Vialette, personal communication).*

We can now define an optimization version of the PIC problem. This is the problem of obtaining one of the compositions which transfers the maximum number of atoms.

PROBLEM 3. MAXIMAL PARTIAL INJECTIONS COMPOSITION *(MPIC)*
INSTANCE*: A collection of n distinct sets $\mathcal{X} = \{X_1, \ldots, X_n\}$, a collection of m partial injections (defined on the sets in $\mathcal{X}$) $\mathcal{I} = \{I_1, \ldots, I_m\}$*
SOLUTION*: A composition of partial injections $I_{comp} \in \mathcal{I}^*$ from X_1 to X_n*
MEASURE*: The size of I_{comp}*

PROPOSITION 2. *The MPIC problem is PSPACE-hard*

PROOF. By proposition 1 the MPIC problem is PSPACE-hard as it is the optimization version of the PIC problem. □

In practice, we are not interested in finding only one solution of MPIC but all of them. We should therefore enumerate all feasible solutions or construct an automaton that accepts all solutions. This automaton is defined in the following way:

DEFINITION 2. *All Maximal Partial Injections Compositions Accepting Automaton (AMPICAA)*
Let $\mathcal{X} = \{X_1, \ldots, X_n\}$ be a collection of n distinct sets and let $\mathcal{I} = \{I_1, \ldots, I_m\}$ be a collection of m partial injections (defined on the sets in $\mathcal{X}$).
We say that the finite state automaton $\mathcal{A} = (Q, \mathcal{I}, \delta, q_{init}, F)$ is an All Maximal Partial Injections Compositions Accepting Automaton (with Q its finite set of states, $\mathcal{I}$ its finite set of symbols, $\delta : Q \times \mathcal{I} \to Q$ its transition function, q_{init} its initial state and $F \subseteq Q$ its set of final states) of $(\mathcal{X}, \mathcal{I})$ iff:
$I_{comp} \in \mathcal{I}^*$ and $\mathcal{A}$ accepts $I_{comp} \Leftrightarrow I_{comp}$ is a maximal composition of partial injections from X_1 to X_n

NOTE 2. *The advantage of building an automaton instead of enumerating all compositions comes from the fact that the number of valid compositions may increase exponentially with the number of states and transitions of the automaton.*

In the rest of this paper, we shall construct a particular instance of AMPICAA, we call AMPICAAβ. An AMPICAAβ is basically an AMPICAA where each state is uniquely labeled by a subset of atoms.

DEFINITION 3. *AMPICAAβ*
Let $\mathcal{A} = (Q, \mathcal{I}, \delta, q_{init}, F)$ be an AMPICAA of $(\mathcal{X}, \mathcal{I})$, let $\tilde{\mathcal{X}}$ be the set of all subsets of all X_i and let $\beta : Q \to \tilde{\mathcal{X}}$ be an injection that associates each state in Q to a unique subset in $\tilde{\mathcal{X}}$, we say that $\mathcal{A}_\beta = \left(Q, \tilde{\mathcal{X}}, \mathcal{I}, \delta, \beta, q_{init}, F\right)$ is the AMPICAAβ of $(\mathcal{X}, \mathcal{I})$ iff:

- $\beta(q_{init}) = X_1$
- $\forall \left(q_i, q_k, I_j\right) \in (Q \times Q \times \mathcal{I}), \delta\left(q_i, I_j\right) = q_k \Rightarrow I_j\left(\beta\left(q_i\right)\right) = \beta\left(q_k\right)$

The β function is very useful to recover the atoms beeing actually transferred from source to any final state of the automata since these atoms are simply given by the label of this state.

In the next paragraph, we give an algorithm to construct the AMPICAAβ automaton.

Computation of the all maximal partial injections compositions accepting sutomaton

The following algorithm (Algorithm 1) is a 'Best First' algorithm to construct the automaton that accepts exactly all words corresponding to maximal compositions of partial injections from X_1 to X_n. The algorithm constructs the automaton by successively adding new states. As we do not know, when adding a new state, if it will be part of the solution, we have to minimize the number of unnecessary states we add to the automaton. A simple heuristic is to select the state associated to the largest subset of atoms (a breadth first implementation of this algorithm is also available).

The lines 21 and 22 in algo. 1 remove the states that have been created but that do not actually belong to the result automaton (Fig. 3 shows these two cases).

Function BEST-FIRST → Automaton
Parameter: SetOfNodes: *initialSet*,
SetOfNodes: *targetSet*,
SetOfPartialInjections: *I*;
Variable: Automaton: *result*,
State: *newState*, *currentState*,
SetOfNodes: *currentSet*, *newSet*,
integer: *minSize*;
begin
1 *minSize* ← 1;
2 *currentState* ← create a state;
3 associate *currentState* with *initialSet* and store it in *result*;
4 label *currentState* as initial;
5 **while** *currentState* ≠ *null* **do**
6 *currentSet* ← the *SetOfNodes* associated to *currentState*;
7 **foreach** *partial injection i* ∈ *I applicable to currentSet* **do**
8 *newSet* ← apply *i* to *currentSet* ;
9 **if** $sizeOf$ (*newSet*) ≥ *minSize* **then**
10 **if** *there exists a state in result associated to newSet* **then**
11 *newState* ← retrieve the state in *result* associated to *newSet*;
12 **else**
13 *newState* ← create a state;
14 associate *newState* with *newSet* and store it in *result*;
15 **if** *newSet* ⊆ *targetSet* **then**
16 label *newState* as final;
17 *minSize* ← max(minSize, sizeOf(*newSet*));
18 create the transition $\delta(currentState, i) = newState$;
19 label *currentState* as explored;
20 *currentState* ← pick the state from *result* not labeled explored or final and associated to the *SetOfNodes* of greatest size (and of size ≥ *minSize*);
21 remove states associated with *SetOfNodes* of size < *minSize* in *result*;
22 remove dangling states in *result*;
23 → *result*;
end

Algorithm 1: The BEST-FIRST algorithm for constructing the automaton accepting all compositions of maximal size between two given sets.

Similarly, by giving the minimum size as a parameter and removing lines 1 and 17, we can directly construct the automaton accepting all the compositions of partial injections of a given minimum size.

In the context of the metabolic pathways reconstruction, we may additionally want to remove two types of states corresponding to two additional constraints on the final automata:

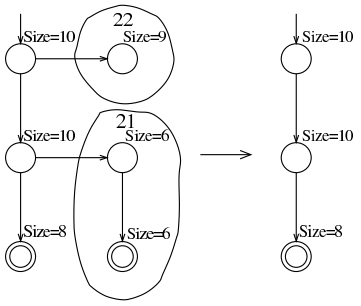


Fig. 3. Illustration of the two types of states removed by lines 21 and 22 in algo. 1.

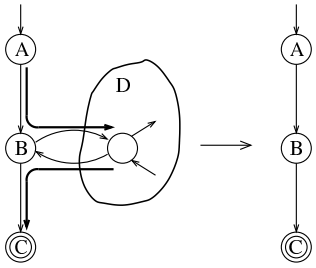


Fig. 4. States removed by the second condition.

- in the automata, there is no limit on the length of words that can be recognized (or, in other terms, in the number of partial injections that may be composed). When we are interested, for biological reason, in words of a limited length (let's say above a given threshold d), then some unnecessary states may be further removed. By doing so, we guarantee that any state kept after this removal is involved in the recognition of at least one word of length d or less and that no word of length d or less is missed.
- we may additionnaly want to remove some special states we called 'dead ends'. As shown in Figure 4, a state D is a 'dead end' iff:
  i) state D can only be reached from a single state B
  ii) any path from D to a final state should go through state B

  'Dead ends' may be safely removed.

These two additional constraints usually greatly reduce the number of states in the final automata as we shall show later.

Table 1. Summary of automata construction for the tryptophan biosynthesis

Initial compound	Final compound	#carbons transferred	#states	#transitions	Computation time
erythrose 4-phosphate	chorismate	4 (max)	5389	2011	1'30"
chorismate	tryptophan	7 (max)	20	48	2"
chorismate	tryptophan	6	87	116	5"

Finally, in the context of metabolic pathways reconstruction, other specific constraints can be added. For instance, instead of specifying the minimum number of all transferred atoms, one may specify a different minimum number for each type of atoms (C, N, O, P...). Or one may restrict the initial and final states to specific atoms (e.g. atoms belonging to a cycle).

APPLICATIONS

Chemical data

The data were extracted from the LIGAND database (Goto *et al.*, 2002) (November 2002 version) which is a part of the KEGG database (Kanehisa *et al.*, 2002). LIGAND contains two types of data: chemical compounds together with their molecular structure and biochemical reactions. All the reactions contained in LIGAND have been pre-processed by a MAXIMUM COMMON SUBGRAPH solving program following the algorithm described in Arita (2000). This resulted in a total of 6191 different partial injections, involving 2920 different compounds and 3737 different reactions. In some applications we further reduce this dataset to the carbon atoms only (this reduces the number of partial injections to 4404, involving 2894 different compounds and 3721 different reactions). In addition, we assumed that all biochemical reactions are reversible and we considered each partial injection twice (i.e. in both directions of the reaction).

Tryptophan biosynthesis

In this paragraph, we give an illustration of our approach to the biosynthesis of tryptophan (one of the three aromatic amino acids together with phenylalanine and tyrosine). These three compounds are synthesized from the same compound called chorismate which is derived from erythrose 4-phosphate.

We applied the algorithm described in the previous section on the database consisting of mappings reduced to carbons atoms in order to construct the following 3 automata:

- the automaton accepting all compositions of maximal size from erythrose 4-phosphate to chorismate;
- the automaton accepting all compositions of maximal size from chorismate to tryptophan;
- the automaton accepting all compositions of size at least 6 from chorismate to tryptophan.

The properties of these 3 automata together with their construction times (PIII processor, Java implementation) are summarized in Table 1. The first automaton (erythrose 4-phosphate to chorismate) displays typical size of automaton (several hundreds of states) and computation time (few minutes).

One can observe that the number of states (and computation time) increases when the specified number of transferred atoms decrease. This comes from the fact that more and more mappings (reactions) get involved when the minimum number of transferred atoms decrease.

Figure 5 displays the automaton accepting all compositions from chorismate to tryptophan with a minimum of 6 carbon atoms transferred and a threshold length $d = 6$. For readability, the states are represented by the complete molecular structure (i.e. not only the subsets of atoms). The transitions are labeled with the KEGG Id of the corresponding reaction. As shown in this figure, the two previously described additional constraints reduce the number of states from 87 to 32 and the number of transitions from 116 to 57. The transitions marked in bold correspond to the reactions contained in the KEGG map entitled 'Tyrosine, phenylalanine and tryptophan biosynthesis". We can see that they form a path from the chorismate to the tryptophan molecule. This path actually corresponds to the known biosynthetic pathway from chorismate to tryptophan. One of the other possible paths in the automaton corresponds to the catabolic pathway leading from tryptophan to chorismate. This is not surprising since we considered all reactions as reversible.

Glycolysis pathway

In practical situations, the number of states can increase dramatically when the number of transferred atoms becomes too small (this comes from the fact that more and more reactions in the database can yield such a transfer). In that case, it may be advantageous to increase the number of transferred atoms by working with all heavy atoms (C, N, P, O) instead of just carbons. This is illustrated by the following example: we ask to convert α-D-glucose 6-phosphate to phosphoenolpyruvate (8 steps out of the 10

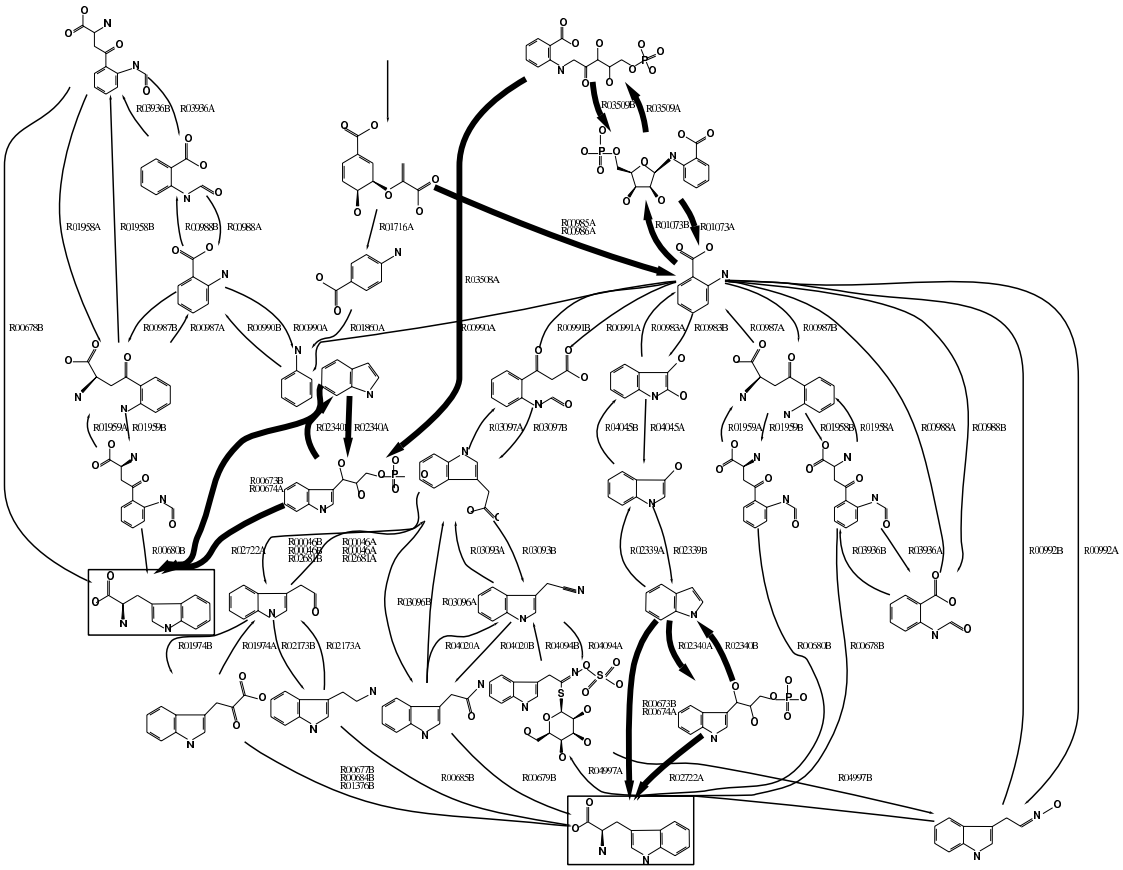


Fig. 5. The automaton accepting all compositions from chorismate to tryptophan with a minimum of 6 carbon atoms transferred and a threshold length of 6. Transitions corresponding to the known biosynthetic pathway from chorismate to tryptophan are drawn in bold.

Table 2. Summary of automata construction for the glycolysis

Initial compound	Final compound	#atoms transferred	threshold length d	#states	#transitions	Computation time
α-D-glucose 6-phosphate	phosphoenolpyruvate	9 (max)	no	686	2368	15'
α-D-glucose 6-phosphate	phosphoenolpyruvate	9 (max)	9	52	161	15'
α-D-glucose 6-phosphate	phosphoenolpyruvate	9 (max)	8	34	96	15'
α-D-glucose 6-phosphate	phosphoenolpyruvate	9 (max)	7	22	58	15'
α-D-glucose 6-phosphate	phosphoenolpyruvate	9 (max)	6	7	9	15'

steps of the glycolysis) by using mappings for all heavy atoms.

We constructed the automaton accepting all compositions of maximal size from α-D-glucose 6-phosphate to phosphoenolpyruvate. We applied the optionnal constraint described in the previous section to limit the length d of the solution. The results are summarized in Table 2. One can observe that this constraint greatly reduces the size of the final automata.

Figure 6 displays the automaton accepting all composi-

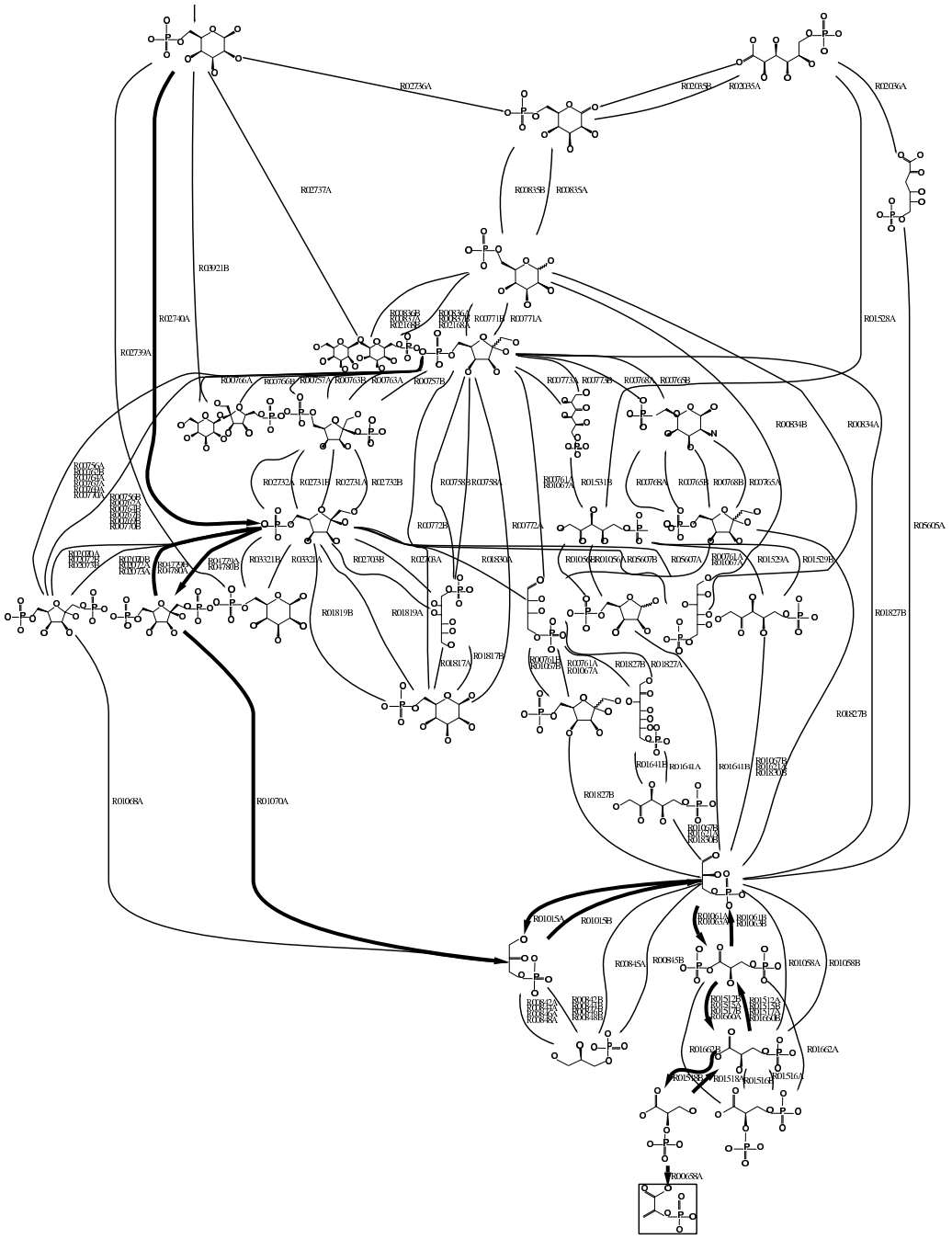


Fig. 6. The automaton accepting all compositions from α-D-glucose 6-phosphate to phosphoenolpyruvate with 9 heavy atoms transferred and a threshold length of 8. Transitions corresponding to reactions involved in the known glycolysis pathway are drawn in bold.

tions from α-D-glucose 6-phosphate to phosphoenolpyruvate with 9 heavy atoms transferred (the maximum) and a threshold length $d = 8$. This automaton contains the 8 steps of the known glycolysis pathway (although one mapping (between fructose 1,6-biphosphate and glyceraldehyde 3-phosphate) is missing).

CONCLUSION AND FUTURE WORK

We show that, despite its bad theoretical computational complexity, this new formulation for the *ab initio* metabolic reconstruction problem, leads to a practical algorithm with reasonable running times and seems to give biologically meaningful and human tractable results. As we mentionned in the introduction, we impose very weak constraints (absence of balance flux conditions, no definition of ubiquitous compounds) to the solutions since the only criterion we use is the number of transferred atoms. This simple criterion may lead to thermodynamically unrealistic solutions. On the other hand, it allows a combinatorial approach of the problem, by contrast to numerical approaches involving thermodynamical criteria. However, it is still possible to post-process the result in order to introduce some thermodynamical scoring scheme.

An interesting extension of this work deals with incomplete databases (i.e. a database with missing reactions). Since our approach makes use of the chemical structures involved, it becomes possible to 'invent' new reactions based on the known ones or on chemical rules. There are mostly two possible extensions in this line. The first one is to infer new reactions based on the structural similarities between substrates and products (i.e. to generate new reactions that have substrates and products similar to the substrates and products of a known reaction). The second one is to substitute compounds (either substrates or products) in a known reaction, for instance to replace a compound by a similar one within a known reaction. An important particular case is the replacement of optical isomers or of labeling errors in the database (the same compound having different labels in different reactions).

ACKNOWLEDGEMENTS

The authors would like to thank Stéphane Vialette and Laurent Trilling for helpful discussions and constructive comments on this work.

REFERENCES

Akutsu,T. (2003) Efficient extraction of mapping rules of atoms from enzymatic reaction data. In *Proceedings of the Conference on Research in Computational Molecular Biology*, pp. 1–8.

Arita,M. (2000) Metabolic reconstruction using shortest paths. *Simulat. Pract. Theory*, **8**, 109–125.

Arita,M. (2000) Graph modeling of metabolism. *J. JPN Soc. Artificial Intelligence*, **15**, 703–710.

Birget,J.C., Margolis,S.W., Meakin,J.C. and Weil,P. (2000) PSPACE-complete problems for subgroups of free groups and inverse finite automata. *Theor. Comput. Sci.*, **242**, 247–281.

Crescenzi,P. and Kann,V. (1998) A compendium of NP optimization problems.http://www.nada.kth.se/~viggo/wwwcompendium/

Gaasterland,T. and Selkov,E. (1995) Reconstruction of metabolic networks using incomplete information. In *Proceedings of the conference on Intelligent Systems for Molecular Biology*. pp. 127–135.

Goto,S., Okuno,Y., Hattori,M., Nishioka,T. and Kanehisa,M. (2002) LIGAND: database of chemical compounds and reactions in biological pathways. *Nucleic Acids Res.*, **30**, 402–404.

Mavrovouniotis,M.L. (1993) Identification of qualitatively feasible metabolic pathways. In *Artificial Intelligence and Molecular Biology*. AAAI Press, Menlo Park, USA.

Overbeek,R., Larsen,N., Maltsev,N., Pusch,G.D. and Selkov,E. (1999) WIT/WIT2: Metabolic reconstruction system. In *Bioinformatics, database and systems*. Kluwer academic Publisher, Boston, USA.

Paley,S. and Karp,P.D. (2002) Evaluation of computational metabolic-pathway predictions for *Helicobacter pylori*. *Bioinformatics*, **18**, 715–724.

Kanehisa,M. (1999) KEGG: From genes to biochemical pathways. In *Bioinformatics, database and systems*. Kluwer academic Publisher, Boston, USA.

Kanehisa,M., Goto,S., Kawashima,S. and Nakaya,A. (2002) The KEGG databases at GenomeNet. *Nucleic Acids Res.*, **30**, 42–46.

Karp,P.D., Krummenacker,M., Paley,S. and Wagg,J. (1999) Integrated pathway-genome databases and their role in drug discovery. *Trends Biotechnol.*, **17**, 275–281.

Kozen,D. (1977) Lower bounds for natural proof systems. pp. 254–266.

Küffner,R., Zimmer,R. and Lengauer,T. (2000) Pathway analysis in metabolic databases via differential metabolic display (DMD). *Bioinformatics*, **16**, 825–836.

Savitch,W.J. (1970) Relationship between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, **4**, 177–192.

Schilling,C.H., Letcher,D. and Palsson,B.O. (2000) Theory for the sytemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *J. Theor. Biol.*, **203**, 229–248.

Schuster,S., Fell,D.A. and Dandekar,T (2000) A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nat. Biotechnol.*, **18**, 326–332.

Vialette,S. Personal communation.

Annexe B

Problème Sous-graphe Induit Commun Connexe Maximal

Problème **16** Sous-graphe Induit Commun Maximal
Données : *deux graphes* $\mathcal{G}_1 = (V_1, E_1)$ *et* $\mathcal{G}_2 = (V_2, E_2)$
Réponse : *deux sous-ensembles* $V_1' \subseteq V_1$ *et* $V_2' \subseteq V_2$ *avec* $|V_1'| = |V_2'|$ *et le sous-graphe de* $\mathcal{G}_1$ *induit par* V_1' *et le sous-graphe de* $\mathcal{G}_2$ *induit par* V_2' *sont isomorphes*
Mesure : $|V_1'|$
Optimisation : max

Un problème dérivé est le problème où le sous-graphe induit est contraint à être connexe.

Problème **17** Sous-graphe Induit Commun Connexe Maximal
Données : *deux graphes* $\mathcal{G}_1 = (V_1, E_1)$ *et* $\mathcal{G}_2 = (V_2, E_2)$
Réponse : *deux sous-ensembles* $V_1' \subseteq V_1$ *et* $V_2' \subseteq V_2$ *avec* $|V_1'| = |V_2'|$ *et le sous-graphe de* $\mathcal{G}_1$ *induit par* V_1' *et le sous-graphe de* $\mathcal{G}_2$ *induit par* V_2' *sont isomorphes et connexes*
Mesure : $|V_1'|$
Optimisation : max

Quelque soit la version du problème (sous-graphes induits connexes ou non connexes), le problème reste $\mathcal{NP}$-difficile [Crescenzi and Kann, 1998] (les problèmes de décision associés se réduisent au problème Clique).

Une méthode générale pour la résolution du problème 16 a été proposée [Levi, 1972]. Elle se décompose en deux étapes :

1. construire un *graphe de correspondance*
2. résoudre le problème Clique Maximale pour le graphe de correspondance

B.1 Construction du graphe de correspondance

Le graphe de correspondance $\mathcal{G}_c = (V_c, E_c)$ est construit à partir des deux graphes $\mathcal{G}_1 = (V_1, E_1)$ et $\mathcal{G}_2 = (V_2, E_2)$ pour lesquels on recherche le plus grand sous-graphe induit commun. Comme les nœuds des graphes $\mathcal{G}_1$ et $\mathcal{G}_2$ sont étiquetés, $\mathcal{G}_c$ est définit de la façon suivante :

- $V_c = \{(v_1, v_2) \in V_1 \times V_2 / \text{étiquette}\,(v_1) = \text{étiquette}\,(v_2)\}$
- $E_c = \left\{ \begin{array}{l} (v_1 = (v_1^1, v_2^1), v_2 = (v_1^2, v_2^2)) \in V_c \times V_c \quad / \\ \qquad ((v_1^1, v_1^2) \in E_1 \wedge (v_2^1, v_2^2) \in E_2) \vee ((v_1^1, v_1^2) \notin E_1 \wedge (v_2^1, v_2^2) \notin E_2) \end{array} \right\}$

Exemple : la figure B.1 montre le graphe de correspondance construit en se basant sur deux graphes dont les nœuds sont étiquetés et qui correspondent à la structure des molécules de pyruvate et de sérine.

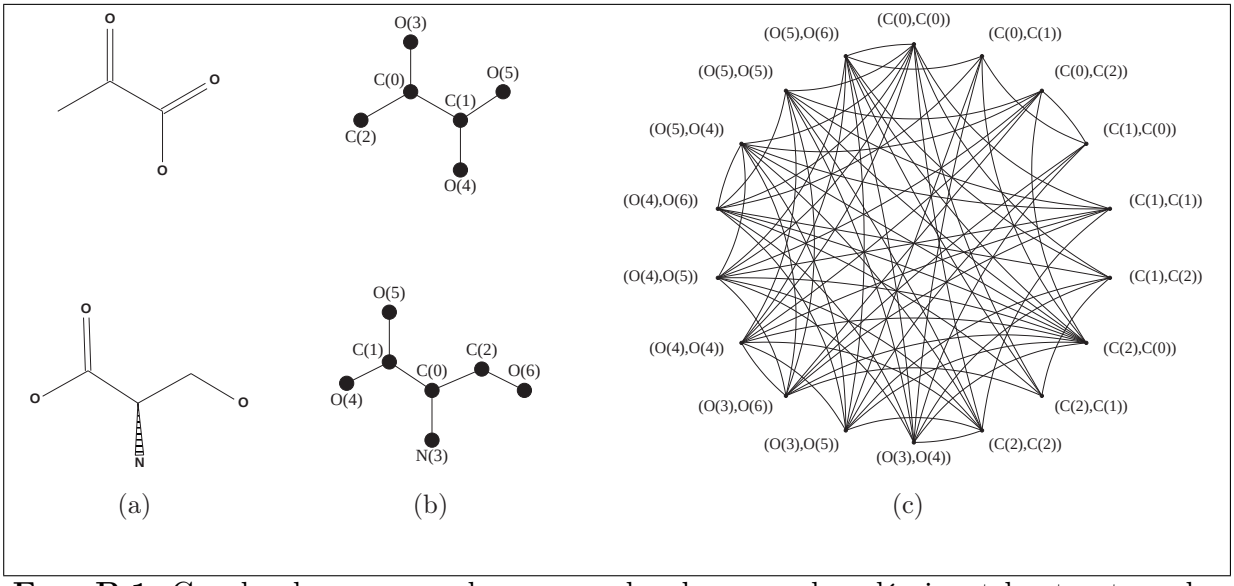


Fig. B.1: Graphe de correspondance pour les deux graphes décrivant la structure des molécules de pyruvate et de sérine - (a) représentation bidimensionnelle des molécules de pyruvate et de sérine, (b) graphes moléculaires associés et (c) graphe de correspondance construit à partir des deux graphes moléculaires

B.2 Résolution du problème Sous-graphe Induit Commun Maximal avec le graphe de correspondance

Chaque clique du graphe de correspondance correspond à une sous-structure conservée et les cliques de taille maximale correspondent aux sous-structures conservées de tailles maximales. Rechercher le sous-graphe induit de taille maximal entre $\mathcal{G}_1$ et $\mathcal{G}_2$ revient donc à rechercher la clique de plus grande taille dans $\mathcal{G}_c$.

Exemple : la figure B.2 montre les sous-structures communes de taille maximale entre les graphes moléculaires du pyruvate et de la sérine et les cliques correspondantes dans le graphe de correspondance pour les deux versions du problème (non-connexe et connexe). Dans le cas de la version connexe du problème, le sous-graphe induit de taille maximale (c) ne correspond pas à une clique de la taille maximale dans le graphe de correspondance (d).

Pour la recherche des cliques dans le graphes de correspondance, il est possible d'utiliser n'importe quel algorithme classique de recherche de clique comme l'algorithme de type Branch&Bound décrit en B.2 (mais il existe bien d'autres algorithmes comme par exemple [Bron and Kerbosch, 1973], voir [Koch, 2001] pour un comparatif de différents algorithmes basés sur la recherche de cliques maximales dans un graphe de correspondance pour rechercher les sous-graphes induits commun à deux ou plusieurs graphes). Il est aussi possible de tenir compte du fait que la recherche des cliques s'effectue dans un graphe de correspondance. En effet, bien que la taille du graphe de correspondance soit de l'ordre de $\mathcal{O}(n^2)$, la taille maximale des cliques est elle bornée supérieurement par le nombre de nœuds du plus petit des deux graphes. Dans le cas où les nœuds des graphes initiaux sont étiquetés (ce qui est le cas ici), il est possible de trouver une borne inférieure encore plus petite que la taille du plus petit des deux graphes. Il est en effet possible de définir à la création du graphe de correspondance des groupes de nœuds qui ne seront jamais inclus dans la même clique car il proposent plusieurs correspondances du même nœud d'un des graphes initiaux (ces groupes sont des *stables* du graphe de correspondance - le graphe de correspondance est un graphe multi-partite). Le nombre de ces groupes constitue donc une borne supérieure à la taille maximale des cliques du graphe de correspondance et ces groupes peuvent être utilisés pour améliorer les performance de la recherche des cliques dans le graphes de correspondance [Durand *et al.*, 1999]. Si on note N_x^1 et N_x^2 le nombre de nœuds associés à l'étiquette x dans les graphe $\mathcal{G}_1$ et $\mathcal{G}_2$, on peut fabriquer un nombre de groupes donné par la formule suivante :

$$\sum_{x \in \{C,O,N,P,Fe,Mg,\ldots\}} \min\left(N_x^1, N_x^2\right)$$

Exemple : la figure B.3 montre les groupes qu'il est possible de constituer pour le graphe de correspondance de la figure B.1.

Il est possible de modifier l'algorithme Branch&Bound de recherche de cliques maximales (algorithme B.2) pour tenir compte de cette contrainte supplémentaire, une adaptation possible est donnée ci après (algorithme B.3).

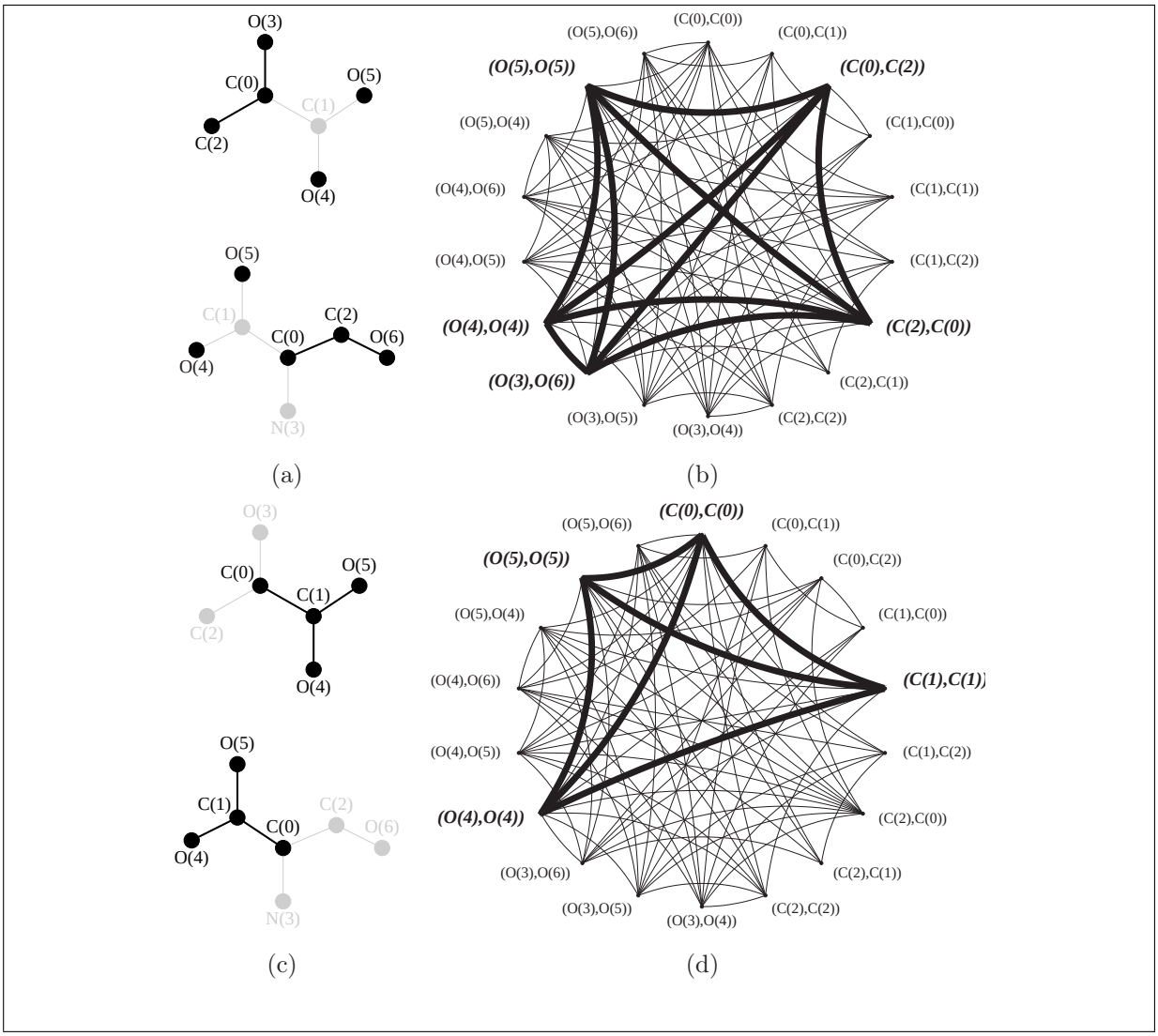


Fig. B.2: Les deux sous-structures entre une molécule de pyruvate et une molécule de sérine correspondant aux solutions du problème Sous-graphe Induit Commun (Connexe) Maximal - (a) montre la plus grande sous-structure commune, de taille 5, entre les deux graphe moléculaires et (b) la clique associée dans le graphe de correspondance, (c) montre la plus grande sous-structure connexe commune qui est elle de taille 4 et (d) la clique associée dans le graphe de correspondance

```
Procédure Branch-and-Bound
Paramètre : Entier : meilleurTaille,
            Ensemble-d-ensembles-de-noeuds : meilleursSolutions,
            Ensemble-de-noeuds : solutionCourante,
            Ensemble-de-noeuds : noeudsCompatibles ;
Variable : Ensemble-de-noeuds : nouveaux,
           Noeud : s ;
début
    si noeudsCompatibles = ∅ alors
        /* Ici, on a nécessairement |solutionCourante| ≥ meilleurTaille */
        /* grâce à la condition d'élagage */
        si |solutionCourante| > meilleurTaille alors
            meilleurTaille ← |solutionCourante|;
            vider(meilleursSolutions);
        meilleursSolutions ← meilleursSolutions ∪ {solutionCourante};
    sinon
        /* On continue l'exploration en profondeur */
        pour chaque s ∈ noeudsCompatibles faire
            noeudsCompatibles ← noeudsCompatibles \ {s};
            nouveaux ← noeuds de noeudsCompatibles reliés à s;
            si (|solutionCourante| + |nouveaux| + 1) ≥ meilleurTaille alors
                Branch-and-Bound(meilleurTaille,
                                 meilleursSolutions,
                                 solutionCourante ∪ {s},
                                 nouveaux);
fin

Programme
Paramètre : Graphe : G = (V, E) ;
Variable : Ensemble-d-ensemble-de-noeuds : meilleursSolutions,
           Ensemble-de-noeuds : noeudsCompatibles;
début
    meilleursSolutions ← ∅;
    noeudsCompatibles ← V ;
    Branch-and-Bound(0, meilleursSolutions, ∅, noeudsCompatibles);
fin Programme
```

Alg. **B.2:** Branch&Bound pour la recherche des cliques

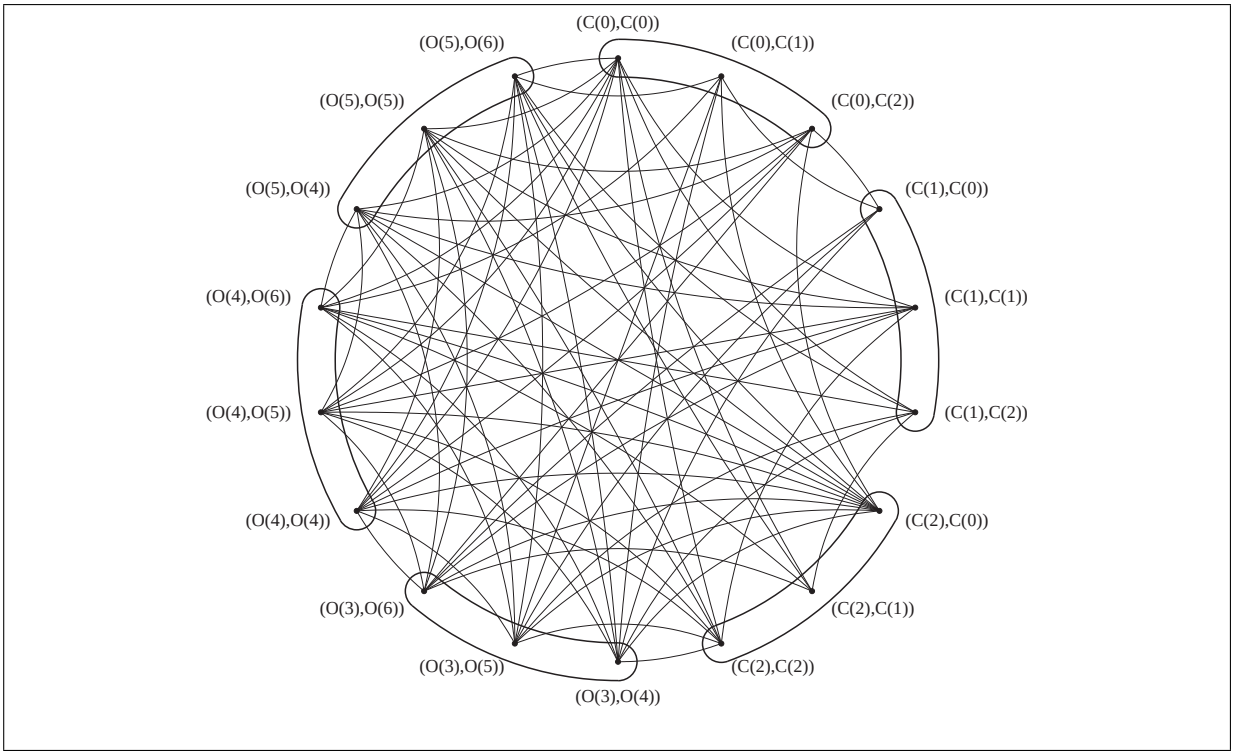


Fig. B.3: Identification de stables dans le graphe de correspondance lors de sa construction - Les nœuds du graphe de correspondance qui sont réunis au sein d'un même groupe proposent tous une correspondance pour un même nœud d'un des deux graphes initiaux et ne peuvent donc pas faire partie de la même solution

```
Procédure B-a-B-Corres
Paramètre : Entier : meilleurTaille,
            Ensemble-d-ensemble-de-noeuds : meilleursSolutions,
            Ensemble-de-noeuds : solutionCourante,
            Ensemble-d-ensemble-de-noeuds : noeudsCompatibles ;
Variable : Ensemble-d-ensemble-de-noeuds : nouveaux,
           Ensemble-de-noeuds : groupeCourant,
           Ensemble-de-noeuds : groupeDeNoeuds,
           Noeud : s ;
début
    si noeudsCompatibles = ∅ alors
        /* Ici, on a nécessairement |solutionCourante| ≥ meilleurTaille */
        /* grâce à la condition d'élagage */
        si |solutionCourante| > meilleurTaille alors
            meilleurTaille ← |solutionCourante|;
            vider(meilleursSolutions);
        meilleursSolutions ← meilleursSolutions ∪ {solutionCourante};
    sinon
        /* On continue l'exploration en profondeur */
        pour chaque groupeDeNoeuds ∈ noeudsCompatibles faire
            pour chaque s ∈ groupeDeNoeuds faire
                groupeDeNoeuds ← groupeDeNoeuds \ {s};
                pour chaque groupeCourant ∈ noeudsCompatibles faire
                    groupeCourant ← noeuds de groupeCourant reliés à s;
                    si groupeCourant ≠ ∅ alors
                        nouveaux ← nouveaux ∪ groupeCourant;
                    si (|solutionCourante| + |nouveaux| + 1) ≥ meilleurTaille alors
                        B-a-B-Corres(meilleurTaille,
                                     meilleursSolutions,
                                     solutionCourante ∪ {s},
                                     nouveaux);
fin

Programme
Paramètre : GrapheDeCorrespondance : G = (V, E),
            Ensemble-d-ensemble-de-noeuds : g ;
            /* g désigne les groupes construits pendant la construction */
            /* du graphe de correspondance */

Variable : Ensemble-d-ensemble-de-noeuds : meilleursSolutions;
début
    meilleursSolutions ← ∅;
    B-a-B-Corres(0, meilleursSolutions, ∅, g);
fin Programme
```

Alg. **B.3:** Branch-and-Bound pour la recherche des cliques de taille maximale dans un graphe de correspondance

B.3 Résolution du problème Sous-graphe Induit Commun Connexe Maximal

Pour résoudre le problème Sous-graphe Induit Commun Connexe Maximal, on ne peut pas utiliser un algorithme classique de recherche de cliques dans le graphe de correspondance car comme illustré sur la figure B.2, la sous-structure connexe de taille maximale ne correspond pas forcément à la clique maximale de taille maximale dans le graphe de correspondance ni même à une clique maximale (ce qui n'est pas le cas dans l'exemple présenté).

Une modification simple consiste à tester lors de l'ajout d'un nouveau nœud à une clique en cours de construction que celle-ci induit toujours un sous-graphe connexe dans les deux graphes initiaux.

Annexe C

Introduction à la notion de complexité

Un moyen de caractériser un problème est de déterminer la 'difficulté intrinsèque' de celui-ci. Par difficulté, on entend, par exemple, le temps de calcul nécessaire à la résolution de ce problème par l'algorithme le plus efficace possible. La théorie de la complexité ne s'intéresse qu'aux problèmes de décision, c'est-à-dire les problèmes où il faut répondre par *oui* ou par *non*.

Exemple : le problème de décision associé à l'atteignabilité d'un nœud d'un graphe depuis un autre nœud de ce graphe est formulé de cette façon :

Problème **18** Atteignabilité
Données : *un graphe* $\mathcal{G} = (V, E)$ *et deux nœuds* $(x, y) \in V^2$
Question : *existe-t-il un chemin de* x *à* y *dans* $\mathcal{G}$ *?*

Le temps de calcul n'est pas la seule ressource à laquelle la notion de complexité est associée, l'autre ressource est l'espace mémoire nécessaire à l'exécution des algorithmes. On parle de complexité en "temps" et en "espace".

C.1 Classe de complexité

Une classe de complexité regroupe des problèmes caractérisés par une série de propriétés communes.

Exemple : la classe de complexité $\mathcal{P}$ correspond aux problèmes pouvant être résolus en temps polynômial.

Le problème Atteignabilité appartient à la classe de complexité $\mathcal{P}$. La preuve de l'appartenance de ce problème à cette classe consiste à exhiber un algorithme dont la complexité est polynômiale et qui résout ce problème. Un parcours en profondeur ou en largeur du graphe permet de résoudre ce problème. L'algorithme en résultant a un temps d'exécution d'au plus n^2 où n est le nombre de nœuds du graphe.

Si tous les algorithmes polynômiaux ne sont pas efficaces en pratique (un algorithme ayant une complexité en n^{100} est inutilisable), il est clair qu'un algorithme ayant une complexité exponentiel en 2^n est forcément inefficace.

Il existe d'autres classes de complexité dont :

- $\mathcal{NP}$: pour tous les problèmes de $\mathcal{NP}$, il est possible de vérifier en temps polynômial si une solution donnée est bonne ou non
- $PSPACE$: les problèmes de $PSPACE$ peuvent tous être résolus avec une taille d'espace mémoire polynômiale en fonction de la taille de l'instance du problème sur une machine de Turing déterministe. Il a été montré que la classe de problèmes $NPSPACE$ (les problèmes peuvant être résolus avec une taille d'espace mémoire polynômiale en fonction de la taille de l'instance du problème sur une machine de Turing non-déterministe) est égale à la classe de problème $PSPACE$ [Savitch, 1970]
- $EXPTIME$: les problèmes de $EXPTIME$ peuvent tous être résolus en temps borné par une exponentielle en fonction de la taille de l'instance du problème

Les propriétés des problèmes de certaines classes de complexité sont redondantes avec les propriétés d'autres classes, par exemple, tout problème résolvable en temps polynômial, qui appartient donc à $\mathcal{P}$, est bien sur résolvable en temps exponentiel et appartient donc à $EXPTIME$.

La structure hiérarchique des classes de complexité présentées est résumée sur la figure C.1.

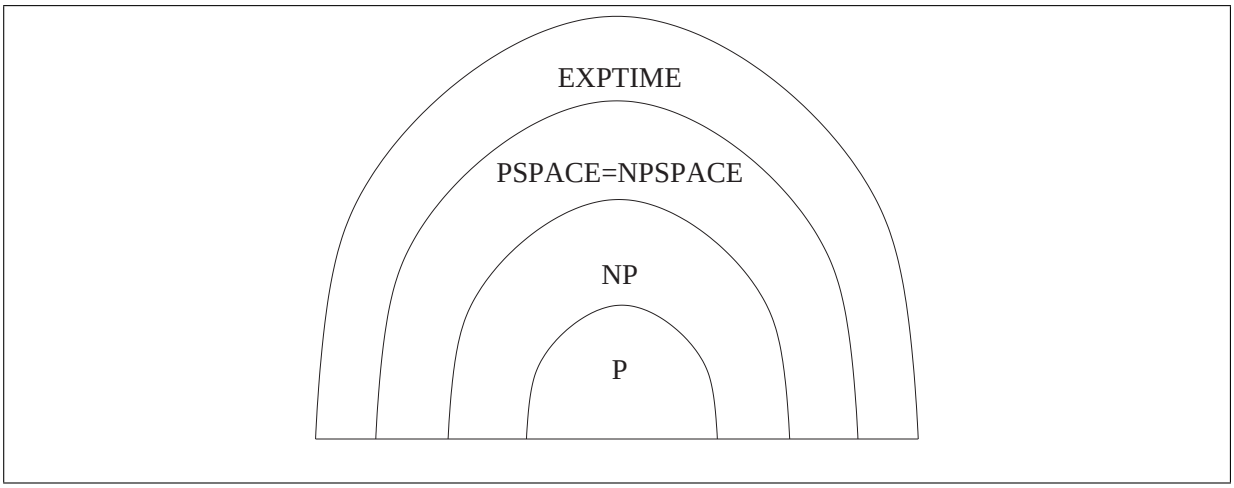


Fig. C.1: Structure des classes de complexité - Cette structure doit toutefois être considérée avec prudence car il n'a jamais été prouvé, même si cela est probable, que l'inclusion de $\mathcal{P}$ dans $\mathcal{NP}$ est stricte

C.2 Problèmes les plus difficiles d'une classe de complexité

Dans les études de complexité, le but est souvent de montrer qu'un problème est 'difficile', c'est-à-dire qu'il fait partie des problèmes les plus difficiles de sa classe de complexité. On parle alors de problème 'complet' pour la classe de complexité.

Exemple : le problème CLIQUE fait partie des problèmes les plus difficiles de $\mathcal{NP}$, on dit donc que CLIQUE est $\mathcal{NP}$-complet.

Les preuves de complexité utilisent la notion de 'réduction' entre problèmes. Pour prouver qu'un problème fait partie des problèmes difficiles de sa classe de complexité, il faut montrer que ce problème est 'au moins aussi difficile' qu'un problème connu pour être lui-même difficile. Le fait que CLIQUE soit $\mathcal{NP}$-complet permet d'affirmer qu'un algorithme capable de résoudre le problème CLIQUE peut être utilisé pour résoudre n'importe quel problème de $\mathcal{NP}$.

Le premier problème de $\mathcal{NP}$ à avoir été prouvé comme étant $\mathcal{NP}$-complet est le problème de la satisfaisabilité d'une équation booléenne.

PROBLÈME **19** SATISFAISABILITÉ
DONNÉES : *un ensemble de clauses* $C = \{C_1, \ldots, C_m\}$ *définies avec les variables booléennes* $\{x_1, \ldots, x_n\}$
QUESTION : *la formule* $C_1 \wedge \ldots \wedge C_m$ *est elle satisfiable (i.e. existe-t-il une assignation de valeurs pour les variables booléennes* $\{x_1, \ldots, x_n\}$ *telle que la formule* $C_1 \wedge \ldots \wedge C_m$ *est vraie) ?*
où chaque clause est un prédicat particulier formé uniquement de la disjonction ($\vee$) *de littéraux (*x *ou* $\neg x$*). Exemple* $C = x \vee \neg y \vee z$.

C.3 Complexité d'un problème d'optimisation

Les problèmes d'optimisation ne sont pas des problèmes de décision, aussi les classes de complexité ne les concernent pas. Il est cependant possible de montrer qu'un problème d'optimisation est 'difficile'. Pour cela on étudie la complexité du problème de décision associé à ce problème d'optimisation. En effet, si il existe un algorithme qui résout le problème d'optimisation, il est possible de l'utiliser (en testant le critère à optimiser) pour résoudre le problème de décision associé.

Si le problème de décision associé est 'complet' pour sa classe de complexité, on dit que le problème d'optimisation est 'difficile'pour cette classe de complexité.

Exemple : le problème d'optimisation CLIQUE MAXIMALE qui consiste à trouver une

clique de taille maximale dans un graphe est $\mathcal{NP}$-difficile, car le problème de décision associé (CLIQUE) est $\mathcal{NP}$-complet.

Bibliographie

[Akutsu, 2003] Tatsuya Akutsu. Efficient extraction of mapping rules of atoms from enzymatic reaction data. In Martin Vingron, Sorin Istrail, Pavel Pevzner, and Michael Waterman, editors, *Proceedings of the International Conference on Research in Computational Biology*, pages 1–8. ACM Press New York, NY, USA, 2003.

[Albert and Barabási, 2002] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74 :47–97, 2002.

[Albert, 2001] Réka Albert. *Statistical mechanics of complex networks.* PhD thesis, University of Notre Dame, Notre Dame, Indiana, USA, april 2001.

[Altshull *et al.*, 1997] Stephen F. Altshull, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped blast and psi-blast : a new generation of protein database searche programs. *Nucleic acids research*, 25(17) :3389–3402, 1997.

[Arita, 2000a] Masanori Arita. Graph modeling of metabolism. *Journal of Japanese Society for Artificial Intelligence*, 15 :703–710, 2000.

[Arita, 2000b] Masanori Arita. Metabolic reconstruction using shortest paths. *Simulation Practice and Theory*, 8 :109–125, 2000.

[Bairoch, 2000] Amos Bairoch. The enzyme database in 2000. *Nucleic acids research*, 28(1) :304–305, 2000.

[Bansal, 2000] Arvind K. Bansal. A framework of automated reconstruction of microbial metabolic pathways. In *Proceedings of the IEEE International Symposium on Bio-informatics and Biomedical Engineering*, 2000.

[Bansal, 2001] Arvind K. Bansal. Integrating co-regulated gene-groups and pair-wise genome comparisons to automate reconstruction of microbial pathways. In *Proceedings of the IEEE International Symposium on Bio-informatics and Biomedical Engineering*, 2001.

[Barabási *et al.*, 1999] Albert-László Barabási, Réka Albert, and Hawoong Jeong. Mean-field theory for scale-free random networks. *Physica A*, 272 :173–187, 1999.

[Barrat and Weigt, 2000] Alain Barrat and Martin Weigt. On the properties of small-world network models. *Eur. Phys. J. B.*, 2000.

[Bateman *et al.*, 2002] Alex Bateman, Ewan Birney, Lorenzo Cerruti, Richard Durbin, Laurence Etwiller, Sean R. Eddy, Sam Griffiths-Jones, Kevin L. Howe, Mhairi Marshall, and Erik L.L. Sonnhammer. The pfam protein families database. *Nucleic acids research*, 30(1) :276–280, 2002.

[Birget *et al.*, 2000] Jean-Camille Birget, Stuart W. Margolis, John C. Meakin, and Pascal Weil. Pspace-complete problems for subgroups of free groups and inverse finite automata. *Theoretical Computer Science*, 242(1–2) :247–281, 2000.

[Bize *et al.*, 2001] Laurent Bize, Daniel Kahn, and Brigitte Mangin. Classification de séquences protéiques : modélisation d'un critère de similitude en utilisant une typologie sur les relations entre protéines. In *Proceedings of the Journées ouvertes de Biologie, Informatique et Mathématiques*, pages 143–150, 2001.

[Blumenthal *et al.*, 2002] Thomas Blumenthal, Donald Evans, Christopher D. Link, Alessandro Guffanti, Daniel Lawson, Jean Thierry-Mieg, Danielle Thierry-Mieg, Wei Lu Chiu, Kyle Duke, Moni Kiraly, and Stuart K. Kim. A global analysis of *caenorhabditis elegans* operons. *Nature*, 417 :851–853, 2002.

[Bollobás, 1985] Béla Bollobás. *Random graphs*. Academic Press, 1985.

[Bono *et al.*, 1998] Hidemasa Bono, Hiroyuki Ogata, Goto Susumu, and Minoru Kanehisa. Reconstruction of amino acid biosynthesis pathways from the complete genome sequence. *Genome research*, 8 :203–210, 1998.

[Brazma *et al.*, 1998] Alvis Brazma, Inge Jonassen, Ingvar Eidhammer, and David Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of computational biology*, 5(2) :279–305, 1998.

[Bron and Kerbosch, 1973] C. Bron and J. Kerbosch. Algorithm 457 - finding all cliques of an undirected graph. *Communications of the ACM*, 16(9) :575–577, 1973.

[Clarke, 1988] Bruce L. Clarke. Stoichiometric network analysis. *Cell biophysics*, 1988.

[Cordwell, 1999] Stuart J. Cordwell. Microbial genomes and "missing" enzymes : redefining biochemical pathways. *Archives of Microbiology*, 172 :269–279, 1999.

[Corpet *et al.*, 2000] Florence Corpet, Florence Servant, Jérome Gouzy, and Daniel Kahn. Prodom and prodom-gc : tools for protein domain analysis and whole genome comparisons. *Nucleic acids research*, 28(1) :267–269, 2000.

[Crescenzi and Kann, 1998] Pierluigi Crescenzi and Vigo Kann. A compendium of np optimization problems, August 1998.

[Dandekar *et al.*, 1999] Thomas Dandekar, Stefan Schuster, Berend Snel, Huynenn Martijn, and Peer Bork. Pathway alignment : application to the comparative analysis of glicolytic enzymes. *Biochem. J.*, 343 :115–124, 1999.

[Dorogovtsev and Mendes, 2001] Sergei D. Dorogovtsev and Jose F.F. Mendes. Evolution of networks. *Adv. Phys*, 2001.

[Durand *et al.*, 1999] Paul J. Durand, Rohit Pasari, Johnnie W. Baker, and Chun-Che Tsai. An efficient algorithm for similarity analysis of molecules. *Internet Journal of Chemistry*, 2 :17, 1999.

[Durbin *et al.*, 1998] Richard Durbin, Eddy Sean, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis : probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[EBI, 2004] EBI. Génomes à l'ebi. http ://www.ebi.ac.uk/genomes/, 2004.

[Eddy, 1998] Sean R. Eddy. Profile hidden markov models. *Bioinformatics*, 14(9) :755–763, 1998.

[Ellis *et al.*, 2001] Lynda B.M. Ellis, C. Douglas Hershberger, Edward M. Bryan, and Lawrence P. Wackett. The university of minnesota biocatalysis/biodegradation database : emphazing enzymes. *Nucleic acids research*, 29(1) :340–343, january 2001.

[Eppstein, 1998] David Eppstein. The k shortest paths problem. *SIAM journal on computing*, 28(2) :652–673, 1998.

[Ermolaeva *et al.*, 2000] Maria D. Ermolaeva, Owen White, and Steven L. Salzberg. Prediction of operons in microbial genomes. *Nucleic acids research*, 29(5) :1216–1221, 2000.

[Falquet *et al.*, 2002] Laurent Falquet, Marco Pagni, Philipp Bucher, Nicolas Hulo, Christian J.A. Sigrist, Kay Hofmann, and Amos Bairoch. The prosite database, its status in 2002. *Nucleic acids research*, 30(1) :235–238, 2002.

[Fan *et al.*, 2002] L.T. Fan, B. Bertók, and F. Friedler. A graph theoretic method to identify candidate mechanisms for deriving the rate law of a catalytic reaction. *Computers and chemistry*, 26 :265–292, 2002.

[Fell and Wagner, 2000] David A. Fell and Andreas Wagner. *BioThermoKinetics 2000 - Animating the cellular map*, chapter 12 Structural properties of metabolic networks : implications for evolution and modelling of metabolism. 2000.

[Fitch, 1970] Walter M. Fitch. Distinguishing homologous from analogous proteins. *Systematic zoology*, 19(2) :99–113, 1970.

[Fitch, 2000] Walter M. Fitch. Homology a personnal view on some of the problems. *Trends in genetics*, 16(5) :227–231, 2000.

[Floyd, 1962] Robert W. Floyd. Algorithm 97 - shortest path. *Communications of the ACM*, 6(5) :345, 1962.

[Fujibuchi *et al.*, 2000] Wataru Fujibuchi, Hiroyuki Ogata, Hideo Matsuda, and Minoru Kanehisa. Automatic detection of conserved gene clusters in multiple genomes by graph comparison and p-quasi grouping. *Nucleic acids research*, 28(20) :4029–4036, 2000.

[Gaasterland and Selkov, 1995] Terry Gaasterland and Evgeni Selkov. Reconstruction of metabolic networks using incomplete information. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, 1995.

[Gagneur *et al.*, 2003] Julien Gagneur, David B. Jackson, and Georg Casari. Hierarchical analysis of dependency in metabolic networks. *Bioinformatics*, 19(8) :1027–1034, 2003.

[Galperin *et al.*, 1998] Michael Y. Galperin, D. Roland Walker, and Eugene V. Koonin. Analogous enzymes : independent inventions in enzyme evolution. *Genome Research*, 8 :779–790, 1998.

[Garey and Johnson, 1979] M.R. Garey and D.S. Johnson. *Computers and intractability : a guide to the theory of NP-completeness.* Freeman, 1979.

[Gattiker *et al.*, 2003] Alexandre Gattiker, Karine Michoud, Catherine Rivoire, Andrea H. Auchincloss, Elisabeth Coudert, Tania Lima, Paul Kersey, Marco Pagni, Christian J.A. Sigrist, Corinne Lachaize, Anne-Lise Veutet, Elisabeth Gasteiger, and Amos Bairoch. Automated annotation of microbial proteomes in swiss-prot. *Computational Biology and Chemistry*, 27 :49–58, 2003.

[Genrich *et al.*, 2001] Hartmann Genrich, Robert Küffner, and Klaus Voss. Executable petri net models for the analysis of metabolic pathways. *Software Tools For Technology Transfer, DOI 10.1007/s100090100058*, 2001.

[Goethals, 2003] Bart Goethals. Survey on frequent pattern mining. Technical report, Department of Computer Science, University of Helsinki, Finland, 2003.

[Goto *et al.*, 2002] Susumu Goto, Yasushi Okuno, Masahiro Hattori, Takaaki Nishioka, and Minoru Kanehisa. Ligand : database of chemical compounds and reactions in biological pathways. *Nucleic acids research*, 30(1) :402–404, 2002.

[Gribskov *et al.*, 1988] M. Gribskov, M. Homyak, J. Edenfield, J Edenfield, and D. Eisenberg. Profile scanning for three-dimensional structural patterns in protein sequences. *Computer applications in the biosciences*, 4(1) :61–66, 1988.

[Habib *et al.*, 2003] M. Habib, C. Paul, and M. Raffinot. Common connected components of interval graphs. Technical Report RR-LIRMM-03014, LIRMM, Université de Montpellier 2, 2003.

[Haft *et al.*, 2001] Daniel H. Haft, Brendan J. Loftus, Delwood L. Richardson, Fan Yang, Jonathan A. Eisen, Ian T. Paulsen, and Owen White. Tigrfams : a protein family resource for the functional identification of proteins. *Nucleic acids research*, 29(1) :41–43, 2001.

[Happel and Sellers, 1989] John Happel and Peter H. Sellers. The caracterization of complex systems of chemical reactions. *Chemical engineering communications*, 83 :221–240, 1989.

[Heiner *et al.*, 2001] Monica Heiner, Ina Koch, and Klaus Voss. Analysis and simulation of steady states in metabolic pathways with petri nets. In *Proceedings of the Workshop on coloured Petri net*, 2001.

[Hofestädt, 2003] Ralf Hofestädt, editor. *In Silico Biology ; Special Issue : Petri Nets for Metabolic Networks*. 2003.

[Huynen *et al.*, 1999] Martijn A. Huynen, Thomas Dandekar, and Peer Bork. Variation and evolution of the citric-acid cycle : a genomic perspective. *Trends in Microbiology*, 7(7) :281–291, july 1999.

[Huynen *et al.*, 2000] Martijn Huynen, Berend Snel, Warren III Lathe, and Peer Bork. Predicting protein function by genomic context : Quantitative evaluation and qualitative inferences. *Genome research*, 10 :1204–1210, 2000.

[Itoh *et al.*, 1999] Takeshi Itoh, Keiko Takemoto, Hirotada Mori, and Takashi Gojobori. Evolutionary instability of operon structures disclosed by sequence comparisons of complete microbial genomes. *Molecular Biology and Evolution*, 16(3) :332–346, 1999.

[Jacob and Monod, 1961] François Jacob and Jacques Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of molecular biology*, 3 :318–356, 1961.

[Jeong *et al.*, 2000] Hawoong Jeong, Bálint Tombor, Zoltán N. Oltvai, and Albert-László Barabási. The large-scale organisation of metabolic networks. *Nature*, 407 :651–653, october 2000.

[Kanehisa and Goto, 2000] Minoru Kanehisa and Susumu Goto. Kegg : Kyoto encycopedia of genes and genomes. *Nucleic acids research*, 28(1) :27–30, january 2000.

[Kanehisa *et al.*, 2002] Minoru Kanehisa, Susumu Goto, Shuichi Kawashima, and Akihiro Nakaya. The kegg databases at genomenet. *Nucleic acids research*, 30(1) :42–46, 2002.

[Karp and Riley, 1999] Peter D. Karp and Monica Riley. *Bioinformatics, database and systems*, chapter 4 EcoCyc : The ressource and the lessons learned. Kluwer academic Publishers, 1999.

[Karp *et al.*, 1972] Richard M. Karp, Raymond E. Miller, and Arnold L. Rosenberg. Rapid identifications of repeated patterns in strings, trees and arrays. In *Proceedings of the ACM symposium on theory of computing*, pages 125–136, 1972.

[Karp *et al.*, 1999] Peter D. Karp, Markus Krummenacker, Suzanne Paley, and Jonathan Wagg. Integrated pathway-genome databases and their role in drug discovery. *Trends in biotechnology*, 17(7), 1999.

[Karp *et al.*, 2000] Peter D. Karp, Monica Riley, Milton Saier, Ian T. Paulsen, Suzanne M. Paley, and Alida Pellegrini-Toole. The ecocyc and metacyc databases. *Nucleic acids research*, 28(1) :56–59, january 2000.

[Karp *et al.*, 2002a] Peter D. Karp, Suzanne M. Paley, and Pedro Romero. The pathway tools software. *Bioinformatics*, 18 Suppl. 1 :S225–S232, 2002.

[Karp *et al.*, 2002b] Peter D. Karp, Monica Riley, Suzanne M. Paley, and Alida Pellegrini-Toole. The metacyc database. *Nucleic acids research*, 30(1) :59–61, 2002.

[Karp *et al.*, 2002c] Peter D. Karp, Monica Riley, Milton Saier, Ian T. Paulsen, Julio Collado-Vides, Suzanne M. Paley, Alida Pellegrini-Toole, César Bonavides-Martínez, and Socorro Gama-Castro. The ecocyc database. *Nucleic acids research*, 30(1) :56–58, 2002.

[Kleinberg, 2000] Jon M. Kleinberg. Navigation in a small world. *Science*, 406 :845, 2000. brief communication.

[Koch, 2001] Ina Koch. Enumarating all connected maximal common subgraphs in two graphs. *Theorical computer science*, 250 :1–30, 2001.

[Kozen, 1977] Dexter Kozen. Lower bounds for natural proof systems. In *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, pages 254–266, 1977.

[Küffner *et al.*, 2000] Robert Küffner, Ralf Zimmer, and Thomas Lengauer. Pathway analysis in metabolic databases via differential metabolic display (dmd). *Bioinformatics*, 16(9) :825–836, 2000.

[Lawrence *et al.*, 1993] Charles E. Lawrence, Stephen F. Altschul, Mark S. Boguski, Jun S. Liu, Andrew F. Neuwald, and John C. Wooton. Detecting subtle sequence signals : a gibbs sampling strategy for multiple alignement. *Science*, 262 :208–213, october 1993.

[Levi, 1972] G Levi. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo*, 9 :341–351, 1972.

[Ma and An-Ping, 2003] Hongwu Ma and Zeng An-Ping. Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics*, 19(2) :270–277, 2003.

[Marais *et al.*, 1999] Armelle Marais, George L. Mendz, Stuart L. Hazell, and Francis Mégraud. Metabolism and genetics of *helicobacter pylori : the genome era. Microbiology and molecular biology reviews*, 63(3) :642–674, 1999.

[Marchler-Bauer *et al.*, 2003] Aron Marchler-Bauer, John B. Anderson, Carol d DeWeese-Scott, Natalie D. Fedorova, Lewis Y. Geer, Siqian He, David I. Hurwitz, John D. Jackson, Aviva R. Jacobs, Christopher J. Lanczycki, Cynthia A. Liebert, Chunlei Liu, Thomas Madej, Gabriele H. Marchler, Raja Mazunder, Anastasiz N. Nikolkaya, Anna R. Pachenko, Bachoti S. Rao, Benjamin A. Shoemaker, Vahan Simonyan, James S. Song, Paul A. Thiessen, Sona Vasudevan, Yanli Wang, Roxanne A. Yamashita, Jodie J. Yin, and Stephen H. Bryant. Cdd : a curated entrez database of conserved domain alignements. *Nucleic acids research*, 31(1) :383–387, 2003.

[Matsuno *et al.*, 2003] H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano. Hybrid petri net representation of gene regulatory network. *In Silico Biology ; Special Issue : Petri Nets for Metabolic Networks*, 2003. http ://www.bioinfo.de/isb/toc_vol_03.html.

[Mavrovouniotis, 1993] Michael L. Mavrovouniotis. *Artificial Intelligence and Molecular Biology*, chapter 9 Identification of qualitatively feasible metabolic pathways. AAAI Press / MIT Press, 1993.

[Mittenthal *et al.*, 1998] Jay E. Mittenthal, Ao Yuan, Bertrand Clarke, and Alexander Scheeline. Designing metabolism : alternative connectivities for the pentose phosphate pathway. *Bulletin of mathematical biology*, 1998.

[Mittenthal *et al.*, 2001] Jay E. Mittenthal, Bertrand Clarke, Thomas G. Waddell, and Glenn Fawcett. A new method for assembling metabolic networks, with application to the krebs citric acid cycle. *Jounal of Theorical Biology*, 208 :361–382, 2001.

[Morgat, 2001] Anne Morgat. Synténies bactériennes. Oral presentation - Entretiens Jacques Cartier on comparative genomics - Lyon, december 2001.

[Mulder *et al.*, 2003] Nicolas J. Mulder, Rolf Apweiler, Teresa K. Attwood, Amos Bairoch, Daniel Barrell, Alex Btemen, David Binns, Margaret Biswas, Paul Bradley, Peer Bork, Phillip Bucher, Richard R. Copley, Emmanuel Courcelle, Ujjwal Das, Richard Durbin, Laurent Falquet, Wolfgang Fleischmann, Sam Griffiths-Jones, Daniel Haft, Nicola Harte, Nicolas Hulo, Daniel Kahn, Alexander Kanapin, Maria Krestyaninova, Rodrigo Lopez, Ivica Letunic, David Lonsdale, Ville Silventoinen, Sandra E. Orchard, Marco Pagni, David Peyruc, Chris P. Ponting, Jeremy D. Selengut, Florence Servant, Christian J.A. Sigrist, Robert Vaughan, and Evgueni M. Zdobnov. The interpro database, 2003 brings increased coverage and new features. *Nucleic acids research*, 31(1) :315–318, 2003.

[Myers, 1996] Eugene W. Myers. Approximate matching of network expressions with spacers. *Journal of Computational Biology*, 3(1) :33–51, 1996.

[Nakaya *et al.*, 2001] Akihiro Nakaya, Susumu Goto, and Minoru Kanehisa. Extraction of

correlated gene clusters by multiple graph comparison. *Genome Informatics*, 12 :44–53, 2001.

[NCBI, 2004] NCBI. Taxonomie au ncbi. http ://www.ncbi.nih.gov/Taxonomy/, 2004.

[Nobeli *et al.*, 2003] Irene Nobeli, Hannes Ponstingl, Eugene B. Krissinel, and Janet M. Thornton. A structure-based anatomy of the e. coli metabolome. *Journal of Molecular Biology*, 334(4) :697–719, 2003.

[Ogata *et al.*, 2000] Hiroyuki Ogata, Wataru Fujibuchi, Susumu Goto, and Minoru Kanehisa. A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic acids research*, 28(20) :4021–4028, 2000.

[Ouzounis and Karp, 2000] Christos A. Ouzounis and Peter D. Karp. Global properties of the metabolic map of *escherichia coli*. *Genome research*, 10 :568–576, 2000.

[Overbeek *et al.*, 1999a] Ross Overbeek, Michael Fonstein, Mark D'Souza, Gordon D. Push, and Natalia Maltsev. The use of gene clusters to infer functional coupling. *Proceedings of the national academy of sciences of the USA*, 96 :2896–2901, march 1999.

[Overbeek *et al.*, 1999b] Ross Overbeek, Niels Larsen, Natalia Maltsev, Gordon D. Pusch, and Evgeni Selkov. *Bioinformatics, database and systems*, chapter 3 WIT/WIT2 : Metabolic reconstruction system. Kluwer academic Publishers, 1999.

[Overbeek *et al.*, 2000] Ross Overbeek, Niels Larsen, Gordon D. Push, Mark D'Souza, Evgeni Jr. Selkov, Nikos Kyripides, Michael Fonstein, Natalia Maltsev, and Evgeni Selkov. Wit : integrated system for high-throughput genome sequence analysis and metabolic reconstruction. *Nucleic acids research*, 28(1) :123–125, 2000.

[Paley and Karp, 2002] Suzanne M. Paley and Peter D. Karp. Evaluation of computational metabolic-pathway predictions for *helicobacter pylori*. *Bioinformatics*, 18 :715–724, 2002.

[Perrière *et al.*, 2000] Guy Perrière, Laurent Duret, and Manolo Gouy. Hobacgen : Database system for comparative genomics in bacteria. *Genome research*, 10 :379–385, 2000.

[Pfeiffer *et al.*, 1999] T. Pfeiffer, I. Sánchez-Valdenebro, J.C. Nuño, F. Montero, and S. Schuster. Metatool : for studying metabolic networks. *Bioinformatics*, 15(3) :251–257, 1999.

[Pisanti and Sagot, 2003] Nadia Pisanti and Marie-France Sagot. *Applied combinatorics on words*, chapter Network expression inference. Cambridge university press, 2003.

[Podani *et al.*, 2001] János Podani, Zoltán N. Oltvai, Hawoong Jeong, Bálint Tombor, Albert-László Barabási, and Szathmáry Eors. Comparable system-level organisation of archaea and eukaryotes. *Nature genetics*, 29(1) :54–56, september 2001.

[Puniyani *et al.*, 2001] Amit R. Puniyani, Rajan M. Lukose, and Bernardo A. Huberman. Intentional walks on scale free small worlds. *cond-mat/0107212*, 2001.

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

[Reddy *et al.*, 1993] Venkatramana N. Reddy, Michael L. Mavrovouniotis, and Michael N. Liebman. Petri nets representations in metabolic pathways. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, 1993.

[Reddy *et al.*, 1996] Venkatramana N. Reddy, Michael N. Liebman, and Michael L. Mavrovouniotis. Qualitative analysis of biochemical reaction systems. *Comput. Biol. Med.*, 26(1) :9–24, 1996.

[Renard-Claudel *et al.*, 2001] Clotilde Renard-Claudel, Claude Chevalet, and Daniel Kahn. Définition de profils spécifiques d'enzymes pour la prédiction des voies métaboliques à partir de génomes complets. In *Proceedings of the Journées ouvertes de Biologie, Informatique et Mathématiques*, pages 239–245, 2001.

[Renard-Claudel *et al.*, 2003] Clotilde Renard-Claudel, Claude Chevalet, and Daniel Kahn. Enzyme-specific profiles for genome annotation : Priam. *Nucleic acids research*, 31(22) :6633–6639, 2003.

[Rison *et al.*, 2002] Stuart C.G. Rison, Sarah A. Teichmann, and Janet M. Thornton. Homology, pathway distance and chromosomal localization of the small molecule metabolism enzymes in *escherichia coli*. *Journal of molecular biology*, 318 :911–932, 2002.

[Salgado *et al.*, 2000] Heledia Salgado, Gabriel Moreno-Hagelsieb, Temple F. Smith, and Julio Collado-Vides. Operons in *escherichia coli* : genomic analyses and predictions. *Proceedings of the national academy of sciences of the USA*, 97(12) :6652–6657, june 2000.

[Salgado *et al.*, 2001] Santos-Zavaleta Alberto Salgado, Heladia, Socorro Gama-Castro, Dulce Millán-Zárate, Edgar Díaz-Peredo, Fabiola Sánchez-Solano, Ernesto Pérez-Rueda, César Bonavides-Martínez, and Julio Collado-Vides. Regulondb (versio 3.2) : transcriptional regulatiojn and operon organization in *escherichia coli* k12. *Nucleic acids research*, 29(1) :72–74, 2001.

[Savitch, 1970] Walter J. Savitch. Relationship between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4 :177–192, 1970.

[Schilling *et al.*, 1999] Christophe H. Schilling, Stefan Schuster, Bernhard O. Palsson, and Reinhart Heinrich. Metabolic pathway analysis, basic concepts and scientific applications in the post-genomic era. *Biotechnol. Prog.*, 15 :296–303, 1999.

[Schilling *et al.*, 2000] Christophe H. Schilling, David Letcher, and Bernhard O. Palsson. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theorical Biology*, 203 :229–248, 2000.

[Schomburg *et al.*, 2002] I. Schomburg, A. Chang, O. Hofmann, C. Ebeling, F. Ehrentreich, and D. Schomburg. Brenda : a resource for enzyme data and metabolic information. *Trends in biotechnology*, 27(1) :54–56, 2002.

[Schuster *et al.*, 2000a] Stefan Schuster, Fell David A., and Thomas Dandekar. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature biotechnology*, 18 :326–332, 2000.

[Schuster *et al.*, 2000b] Stefan Schuster, Thomas Pfeiffer, Ferdinand Moldenhauer, Ina Koch, and Thomas Dandekar. Structural analysis of metabolic networks : elementary flux modes, analogy to petri nets, and application to *mycoplasma pneumoniae*. In *Proceedings of the German Conference on Bioinformatic*, pages 115–120, 2000.

[Schuster *et al.*, 2002] Stefan Schuster, C. Hilgetag, J.H. Woods, and Fell David A. Reaction routes in biochemical reactions systems : Algebraic properties, validated calculation procedure and example from nucleotide metabolism. *Mathematical Biology*, 45 :153–181, 2002.

[Selkov *et al.*, 1997] Evgeni Selkov, Natalia Maltsev, Gary J. Olsen, Ross Overbeek, and William B. Whitman. A reconstruction of the metabolism of *methanococcus jannaschii* from sequence data. *Gene*, 1997.

[Selkov *et al.*, 2000] Evgeni Selkov, Ross Overbeek, Yakov Kogan, Lien Chu, Veronika Vonstein, David Holmes, Simon Silver, Robert Haselkorn, and Michael Fonstein. Functional analysis of gapped microbial genomes : amino acid metabolism of *thiobacillus ferrooxidans*. *Proceedings of the national academy of sciences of the USA*, 97(7) :3509–3514, 2000.

[Seo *et al.*, 2001] H. Seo, D-Y. Lee, S. Park, L.T. Fan, S. Shafie, B. Bertók, and F. Friedler. Graph-theorical identification of pathways for biochemical reactions. *Biotechnology letters*, 23 :1551–1557, 2001.

[Seressiotis and Bailey, 1988] Alex Seressiotis and James E. Bailey. Mps : an artificial intelligent software system for the analysis and synthesis of metabolic pathways. *Biotechnology and bioingineering*, 31 :587–602, 1988.

[Susumu *et al.*, 1997] Goto Susumu, Hidemasa Bono, Hiroyuki Ogata, W. Fujibuchi, T. Nishioka, K. Sato, and Minoru Kanehisa. Organizing and computing metabolic pathway data in terms of binary relations. In *Proceedings of the Pacific Symposium of Bioinformatics*, volume 2, pages 175–186, 1997.

[Tatusov *et al.*, 1996] R.L. Tatusov, A. Mushegian, P. Bork, N.P. Brown, W.S. Hayes, M. Borodovsky, K.E. Rudd, and E.V. Koonin. Metabolism and evolution of *haemophilus influenzae* deduced from a whole-genome comparison with *escherichia coli*. *Current biology*, 6 :279–291, 1996.

[Tatusov *et al.*, 1997] Roman L. Tatusov, Eugene V. Koonin, and David J. Lipman. A genomic perspective on protein families. *Science*, 278 :631–637, october 1997.

[Valette, 2000] Robert Valette. Les réseaux de petri. http ://www.laas.fr/∼robert/ - Support de cours, septembre 2000.

[van Helden *et al.*, 2002] Jacques van Helden, Lorenz Wernisch, David Gilbert, and Shoshana Wodak. *Bioinformatics and genome analysis*, chapter Graph-based analysis of metabolic networks, pages 245–274. Springer-Verlag, 2002.

[Voet and Voet, 1998] Donald Voet and Judith G. Voet. *Biochimie*. DeBoeck Université, 1998.

[Wagner and Fell, 2001] Andreas Wagner and David A. Fell. The small world inside large metabolic networks. *Proc. Roy. Soc. B*, 2001.

[Walsh, 1999] Toby Walsh. Search in small world. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1999.

[Watts and Strogatz, 1998] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393 :440–442, june 1998.

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

[Yada *et al.*, 1999] Tetsushi Yada, Mitsuteru Nakao, Yasushi Totoki, and Kenta Nakai. Modeling and predicting transcriptional units of *escherichia coli* genes using hidden markov models. *Bioinformatics*, 15(12) :987–993, 1999.

[Yanai and DeLisi, 2002] Itai Yanai and Charles DeLisi. The society of genes : networks of functional links between genes from comparative genomics. *Genome Biology*, 3(11), 2002.

[Zheng *et al.*, 2002] Yu Zheng, Joseph D. Szustakowski, Lance Fortnow, Richard J. Roberts, and Simon Kasif. Computational identification of operons in microbial genomes. *Genome research*, 12 :1221–1230, 2002.

Résumé

La reconstruction des voies métaboliques d'un organisme est une tâche importante en biologie et plusieurs approches ont déjà été proposées pour assister ce travail mais il y a un besoin pour des approches plus exploratoires.

La première partie de cette thèse s'intéresse à la reconstruction *ab initio* de voies métaboliques. Cela consiste à retrouver au sein du réseau de l'ensemble des réactions chimiques décrites pour un organisme vivant, un sous réseau connectant au moins deux composés. Nous proposons une nouvelle formulation de ce problème qui considère les réactions comme des transferts d'atomes entre composés chimiques. Une voie métabolique est ainsi associée à un transfert d'atomes entre deux composés. Le problème de la reconstruction est alors de rechercher la succession de réactions maximisant le nombre d'atomes transférés entre ces deux composés. Ce problème est exprimé comme la recherche d'une composition d'injections partielles dont la taille de l'image est maximale. La complexité de ce problème a été étudiée et un algorithme le résolvant est présenté.

La seconde partie présente la formalisation d'un problème de comparaison de graphes. Le cas particulier traité dans cette thèse concerne la comparaison d'un réseau de réactions avec l'organisation spatiale des gènes sur le génome. Cette comparaison permet l'identification de voies métaboliques codées en opérons dans les génomes bactériens.

Mots-clés: bioinformatique, voies métaboliques, reconstruction