Analysis of biochemical reaction graph : application to heterotrophic plant cell metabolism

Vu Ngoc Tung Nguyen

université de BORDEAUX

Année: 2015

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE DE BORDEAUX
SPÉCIALITÉ : INFORMATIQUE

Par

Vu Ngoc Tung NGUYEN

Analyse des graphes de reactions biochimiques avec une application au réseau metabolique de la cellule de plante

Soutenue le 3 février 2015

Membres du jury :

Mme. Anne SIEGEL	DR	Université Rennes 1	Rapporteur
M. Jérémie BOURDON	HDR	Université de Nantes	Rapporteur
Mme. Marie BEURTON-AIMAR	HDR	Université de Bordeaux	Directeur de thèse
M. Pascal DESBARATS	Professeur	Université de Bordeaux	Examinateur
M. Fabien JOURDAN	Docteur	UMR 1331 INRA Toxalim	Examinateur
M. Dominique ROLIN	Professeur	UMR 1332 INRA BP 81	Président du jury
Mme. Sophie COLOMBIÉ	Docteur	UMR 1332 INRA BP 81	Co-encadrante

University of Bordeaux
Doctoral school of Mathematics and Computer Science

Year 2015

PhD Thesis

Analysis of biochemical reaction graph: application to heterotrophic plant cell metabolism

presented by

NGUYEN Vu Ngoc Tung

to receive the Doctoral Diploma of Computer Science

Defended in February 3^{rd}, 2015

Committee in charge :

Mme. Anne SIEGEL	Research Director	University of Rennes 1	Reviewer
M. Jérémie BOURDON	Associate Professor	University of Nantes	Reviewer
Mme. Marie BEURTON-AIMAR	Associate Professor	University of Bordeaux	Supervisor
M. Pascal DESBARATS	Professor	University of Bordeaux	Examiner
M. Fabien JOURDAN	Doctor	UMR 1331 INRA Toxalim	Examiner
M. Dominique ROLIN	Professor	UMR 1332 INRA BP 81	President of the jury
Mme. Sophie COLOMBIÉ	Doctor	UMR 1332 INRA BP 81	Co-supervisor

Dedication

This dissertation is dedicated to my beloved parents, my parents-in-law, my elder brothers, my younger sister, my younger-brother-in-law and my petite family (my loving wife NGO Thi My Hue and my child NGUYEN Ngoc Minh Khang) who have been a source of encouragement and inspiration to me throughout my life.

Acknowledgements

This thesis has been a real journey and I am grateful for the people that I have met and those that I have come to know along the way. There are certain people who have played a big part in the completion of this voyage, and to whom I am very grateful and would like to acknowledge.

I would like to thank my supervisor, Assoc. Prof. Beurton-Aimar Marie, for her support, the invaluable ideas and discussions during the past three years. These have played an important role in the research. Her unfailing patience and support, especially during those times that I have had to spend on other work beside my thesis, is very much appreciated.

I also wish to thank my co-supervisor, Dr. Colombié Sophie, for her helpful advice and for the valuable comments and suggestions that she provided, especially on the explanations of the obtained results.

I would like to thank Vietnamese Ministry of Education and Training (MOET), along with French Ministry of Higher Education and Research, for funding four years of my studies.

I would like to express my sincere gratitude to my parents and my parents in law for their extraordinary support towards my studies. They have helped me throughout my education and without their support this journey would not have been possible.

This list is by no means complete. There are many others who have helped and supported me during my studies and my stay in LaBRI. I would like to convey my sincere thanks to all individuals who have helped me directly or indirectly during my doctoral studies.

Last but not least, I would especially like to thank my wife, Hue, and our child, Tony who have supported me through all far-house years of study. I am grateful for their dependable love and for the inspiration they have given, and continue to give, me. This thesis is dedicated to them.

Nguyen Vu Ngoc Tung
33400 Talence, France
Feb. 2015

Abstract

Nowadays, systems biology are facing the challenges of analysing the huge amount of biological data and large-scale metabolic networks. Although several methods have been developed in recent years to solve this problem, it is existing hardness in studying these data and interpreting the obtained results comprehensively. This thesis focuses on analysis of structural properties, computation of elementary flux modes and determination of minimal cut sets of the heterotrophic plant cellmetabolic network. In our research, we have collaborated with biologists to reconstruct a mid-size metabolic network of this heterotrophic plant cell. This network contains about 90 nodes and 150 edges. First step, we have done the analysis of structural properties by using graph theory measures, with the aim of finding its owned organisation. The central points or hub reactions found in this step do not explain clearly the network structure. The small-world or scale-free attributes have been investigated, but they do not give more useful information. In the second step, one of the promising analysis methods, named elementary flux modes, gives a large number of solutions, around hundreds of thousands of feasible metabolic pathways that is difficult to handle them manually. In the third step, minimal cut sets computation, a dual approach of elementary flux modes, has been used to enumerate all minimal and unique sets of reactions stopping the feasible pathways found in the previous step. The number of minimal cut sets has a decreasing trend in large-scale networks in the case of growing the network size. We have also combined elementary flux modes analysis and minimal cut sets computation to find the relationship among the two sets of results. The findings reveal the importance of minimal cut sets in use of seeking the hierarchical structure of this network through elementary flux modes. We have set up the circumstance that what will be happened if glucose entry is absent. Bi analysis of small minimal cut sets we have been able to found set of reactions which has to be present to produce the different sugars or metabolites of interest in absence of glucose entry. Minimal cut sets of size 2 have been used to identify 8 reactions which play the role of the skeleton/core of our network. In addition to these first results, by using minimal cut sets of size 3, we have pointed out five reactions as the starting point of creating a new branch in creation of feasible pathways. These 13 reactions create a hierarchical classification of elementary flux modes set. It helps us understanding more clearly the production of metabolites of interest inside the plant cell metabolism.

Keywords: Metabolic networks, Elementary Flux Modes, Minimal Cut Sets, Graph-based analysis, Complex networks, Systems biology, Plant cell metabolism
Discipline: Computer Science/Bioinformatics

Résumé

Aujourd'hui, la biologie des systèmes est confrontée aux défis de l'analyse de l'énorme quantité de données biologiques et à la taille des réseaux métaboliques pour des analyses à grande échelle. Bien que plusieurs méthodes aient été développées au cours des dernières années pour résoudre ce problème, ce sujet reste un domaine de recherche en plein essor. Cette thèse se concentre sur l'analyse des propriétés structurales, le calul des modes élémentaires de flux et la détermination d'ensembles de coupe minimales du graphe formé par ces réseaux. Dans notre recherche, nous avons collaboré avec des biologistes pour reconstruire un réseau métabolique de taille moyenne du métabolisme cellulaire de la plante, environ 90 nœuds et 150 arêtes. En premier lieu, nous avons fait l'analyse des propriétés structurelles du réseau dans le but de trouver son organisation. Les réactions points centraux de ce réseau trouvés dans cette étape n'expliquent pas clairement la structure du réseau. Les mesures classiques de propriétés des graphes ne donnent pas plus d'informations utiles. En deuxième lieu, nous avons calculé les modes élémentaires de flux qui permettent de trouver les chemins uniques et minimaux dans un réseau métabolique, cette méthode donne un grand nombre de solutions, autour des centaines de milliers de voies métaboliques possibles qu'il est difficile de gérer manuellement. Enfin, les coupes minimales de graphe, ont été utilisés pour énumérer tous les ensembles minimaux et uniques des réactions qui stoppent les voies possibles trouvées à la précédente étape. Le nombre de coupes minimales a une tendance à ne pas croître exponentiellement avec la taille du réseau a contrario des modes élémentaires de flux. Nous avons combiné l'analyse de ces modes et les ensembles de coupe pour améliorer l'analyse du réseau. Les résultats montrent l'importance d'ensembles de coupe pour la recherche de la structure hiérarchique du réseau à travers modes de flux élémentaires. Nous avons étudié un cas particulier : qu'arrive-t-il si on stoppe l'entrée de glucose ? En utilisant les coupes minimales de taille deux, huit réactions ont toujours été trouvés dans les modes élémentaires qui permettent la production des différents sucres et métabolites d'intérêt au cas où le glucose est arrêté. Ces huit réactions jouent le rôle du squelette / cœur de notre réseau. En élargissant notre analyse aux coupes minimales de taille 3, nous avons identifié cinq réactions comme point de branchement entre différent modes. Ces 13 réactions créent une classification hiérarchique des modes de flux élémentaires fixés et nous ont permis de réduire considérablement le nombre de cas à étudier (approximativement divisé par 10) dans l'analyse des chemins réalisables dans le réseau métabolic. La combinaison de ces deux outils nous a permis d'approcher plus efficacement l'étude de la production des différents métabolites d'intérêt par la cellule de plante hétérotrophique.

Mots-clef: Réseaux métaboliques, Modes élémentaires, Minimal Cut Sets, Graph-based analysis, réseaux complexes, biologie systémique, métabolisme des plantes

Discipline: Informatique/Bioinformatique

Introduction

Un réseau métabolique est constitué d'un ensemble de réactions (équations) qui décrivent une suite de transformations biochimiques. Jusque très récemment, l'échelle des réseaux étudiés se situait au niveau d'une voie métabolique. Bien que certaines voies puissent être relativement complexes, de l'ordre d'une dizaine de réactions impliquées, le raisonnement conduit pour leur analyse, se basait sur des algorithmes supposant un comportement `"linéaire"`, c'est à dire que les cycles étaient éliminés et que lorsque deux voies, deux branches, étaient possibles, chacune était analysée séparément. Dès que les biologistes ont désiré réaliser ces analyses à l'échelle d'un organisme (ou d'un organelle) il est devenu indispensable de repenser les méthodes et plus encore les outils pour conduire ces analyses. En effet ce changement d'échelle provoque un changement drastique du niveau de complexité du réseau étudié et pas seulement un accroissement quantitatif du nombre de réactions à analyser. Un réseau, quel qu'en soit sa nature - réseau social, routier, grille de processeur, processus industriels, etc, peut-être modélisé par un graphe, orienté ou non. Les outils mathématiques ou informatiques dédiés aux graphes sont donc utilisables pour modéliser et analyser les réseaux biologiques.

Dans cette thèse, nous décrirons dans un premier temps les spécificités des réseaux métaboliques et le type de graphe adéquat à leur modélisation. Puis nous étudierons les différentes formalisations des graphes d'interactions et nous montrerons que la méthode des modes élémentaires de flux est un outil puissant pour analyser ces graphes à l'échelle des systèmes. Nous aborderons également les ensembles de coupes minimales, outils complémentaires aux modes élémentaires de flux. La dernière partie de cette thèse sera consacrée à une extension de cette méthode que nous proposons. Cette extension nous permet de définir des modes élémentaires de métabolites. Toutes les méthodes ont été utilisées sur plusieurs réseaux métaboliques, 3 réseaux qui modélisent le métabolisme mitonchondrial dans différents tissus : muscle, foie et levure, et un réseau qui modélise le métabolisme central carboné des plantes. Pour cet exemple, nous déclinerons plusieurs situations suivant les différentes productions de sucre ou d'acides aminées qui ont été étudiées.

Description du graphe d'interactions

Traditionnellement, l'analyse d'un réseau métabolique consiste à réunir un ensemble de réactions de la forme :

$$Reaction\_i : substrat1 + substrat2 = produit1 + produit2$$

Cette réaction décrit la transformation biochimique des deux métabolites *substrat*1 et *substrat*2 en deux autres métabolites *produit*1 et *produit*2. On peut associer un nom à cette réaction, la description du réseau sera donc une liste de réactions similaires à celle ci-dessous.

Nom Réaction	Substrats		Produits
Glucokinase :	Glucose + ATP	=	Glucose-6P + ADP
Isomerase :	Glucose-6P	=	Fructose-6P
Fructokinase :	Fructose-6P + ATP	=	Fructose-6biPhosphate + ADP

Puisque l'ensemble des réactions à l'échelle d'un organisme peut être très grand, on décompose cet ensemble en unité fonctionnelle appelée *voie métabolique*. Cette décomposition, parfois arbitraire, fait appel au concept de fonction biologique. Pour simplifier, on peut définir une fonction biologique comme un ensemble ordonné de réactions concourant à un même objectif. Par exemple la production de sucre (glucose) pour la *glycolyse*.

Réseau et graphe : L'outil naturel en informatique pour représenter des interactions entre différents éléments est le graphe. Un graphe est défini par un ensemble fini de sommets ou noeuds V (ou vertices) et un ensemble E d'arêtes (ou edges) avec $E \subseteq VxV$. Les arêtes représentent les

relations entre les sommets ; les arêtes et les sommets peuvent être étiquetés. Les arêtes peuvent également être valuées, on parlera alors de poids. Un graphe peut être orienté ou non et supporter plusieurs types de sommets. La question de représenter un réseau biologique par un graphe pose la question du choix des entités biologiques qui seront associées aux sommets et aux arêtes. Dans le cadre du métabolisme, il existe plusieurs possibilités. Les sommets peuvent être les réactions, on parlera alors de graphes de réactions, ou bien les métabolites, nommé dans ce cas graphes de métabolites [1], c'est la représentation classique que l'on peut trouver dans la littérature en biologie. On peut aussi créer un graphe appelé bi-partie avec deux types de sommets, les métabolites et les réactions. Lorsque les sommets représentent uniquement des métabolites, les réactions sont positionées sur les arêtes, c'est la représentation choisie dans la figure 1. Comme on peut le voir dans cette figure, dès que la réaction a plus d'un substrat et un produit, une situation très fréquente, le graphe généré est appelé *hypergraphe*. Si cette structure est aisément compréhensible visuellement, son traitement par des méthodes algorithmiques de la théorie des graphes est plus complexe, aussi on traduira le plus souvent un *hypergraphe* par un graphe bi-partie, explicitant l'association de plusieurs substrats dans une réaction ou la génération de plusieurs produits.

C'est le choix qui a été fait par les différents projets internationaux de représentation de connaissances sur les réseaux métaboliques comme KEGG (Kyoto Encyclopedia of Genes and Genomes) ou MetaCyc (Encyclopedia of Metabolic Pathway). La figure 1 montre à nouveau la chaîne de la glycolyse telle qu'elle apparaît sur le site de KEGG, les réactions sont les noeuds rectangulaires, les noms des réactions sont insérés dans ces rectangles, les métabolites sont symbolisés par les petits noeuds ronds, leur nom est inscrit à coté de ce rond. Les flêches sur les arêtes permettent de spécifier la réversibilité des réactions, information importante pour comprendre le jeu de contraintes qui s'exercent sur les interactions.

Graphes bi-partie : Un réseau de Petri [2] est un modèle bien connu en informatique de graphe bi-partie qui permet la simulation du fonctionnement d'un réseau sur un modèle de production/consommation. Plusieurs auteurs [3, 4] ont montré l'intérêt de cet outil pour la modélisation des réseaux métaboliques car un élément important de la définition de ces réseaux est qu'ils décrivent la consommation de molécules (les substrats) et la production de nouvelles molécules (les produits) qui deviendront à leur tour les substrats d'autres réactions. Les réseaux de Petri sont donc particulièrement adaptés pour représenter ces phénomènes surtout lorsqu'on désire simuler le fonctionnement d'une ou plusieurs voies métaboliques interagissant et mises en concurrence pour l'utilisation de molécules communes. Malgré ces avantages, ce n'est pas l'outil que nous avons retenu pour nos études car ainsi que nous l'avons dit, les réseaux de Petri sont utilisés en simulation et notre travail sur l'analyse des réseaux métaboliques concernent plutôt les aspects statiques : structure, topologie pour lesquels les réseaux de Petri ne sont pas obligatoirement les plus adaptés. Toutefois, nous verrons qu'il existe des liens forts entre les outils que nous avons utilisés, les modes élémentaires de flux, et certaines propriétés des réseaux de Petri.

Complexité

Un des éléments fondamentaux de la complexité d'un réseau biologique est la concurrence à laquelle se livrent différentes réactions pour consommer le même métabolite mais aussi le fait que le même métabolite peut être produit par différentes réactions. Une première approche de la mesure de cette complexité peut être obtenue par différents éléments de cardinalité des noeuds comme le nombre de substrats/produits participant à une réaction donnée ou bien, le nombre de réactions différentes reliées au même métabolite. Si l'on considère un réseau métabolique comme une graphe bi-partie, c.-à-d. ayant deux types de nœuds, l'arité moyenne suivant les types est un bon indicateur de la différence de complexité percue intuitivement, suivant qu'on considère le réseau des réactions ou des métabolites. Bien qu'il n'existe pas de règle sur le nombre de métabolites impliquées, substrats ou produits, par réaction, l'expérience montre que le plus souvent l'ordre de grandeur du nombre de molécules impliquées se situe entre 2 et 5/6. L'arité moyenne des nœuds réactions varie donc

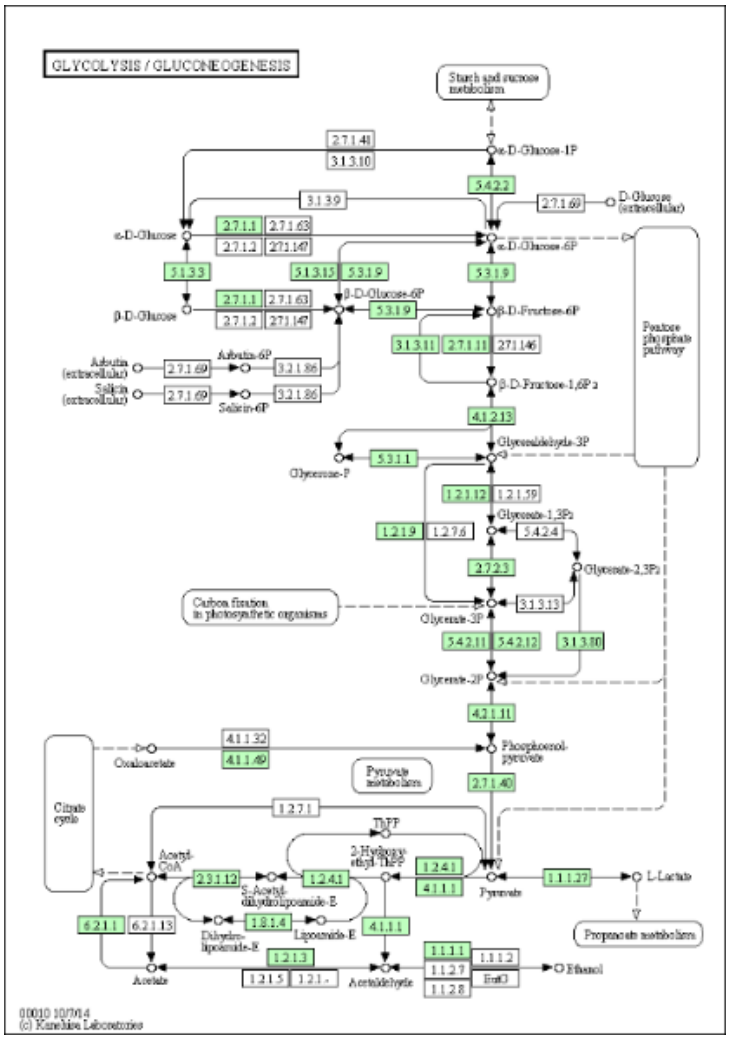


FIGURE 1 – Chaîne de la glycolyse dans la base de donnée KEGG

peu et dans nos exemples de réseaux, on peut constater que l'arité moyenne des nœuds réactions est indépendante de la taille du réseau. Il en est tout autre pour l'arité des nœuds métabolites qui peut se révéler drastiquement différente de celle des nœuds réactions. Ces métabolites fortement utilisés dans le réseau sont généralement appelés métabolites *"hubs"* en ceci qu'ils deviennent des incontournables au moment de calculer le comportement du système.

Les modes élémentaires de flux

Les premiers travaux de notre équipe sur l'utilisation des modes élémentaires de flux (*efms*) dans le cadre de l'étude du métabolisme énergétique de la mitochondrie ont fait l'objet de la thèse de Sabine Pérès. Actuellement, nous nous focalisons sur l'étude du métabolisme carboné de la plante.

La méthode d'identification des modes élémentaires de flux d'un réseau métabolique consiste à déterminer les voies métaboliques admissibles de ce réseau à partir de sa matrice de stochiométrie. Les seules informations utilisées par cette méthode sont la topologie du réseau (coefficient de stochiométrie, réversibilité/irréversibilité des réactions) et ne nécessite pas de connaissance des paramètres cinétiques des réactions. On retiendra comme principe de base de cette méthode qu'elle détermine **les chemins uniques et minimaux** du graphe en respectant la contrainte que le réseau métabolique doit être à l'état stationnaire. Cette analyse topologique permet de caractériser des propriétés du réseau comme la robustesse du réseau (ou son niveau de redondance) [5], les réactions qui opèrent toujours (ou jamais) ensemble...La recherche de voies métaboliques ou suites de réactions correspondant à une fonction biologique a longtemps été considéré comme triviale dans la mesure ou les voies considérées correspondaient aux ensembles de réactions (le plus souvent de l'ordre d'une dizaine de réactions) bien connus dans la littérature. Le passage à l'échelle du système oblige à considérer désormais des ensembles pouvant aller jusqu'à plusieurs centaines de réactions. Ceci conduit inéluctablement à la production de plusieurs milliers de solutions. Stelling et al. [5] ou Wilhelm et al. [6] ont étudié les conséquences de tels résultats en terme de mesure de robustesse des réseaux et apporté un nouvel éclairage sur la façon de considérer la robustesse des fonctions biologiques.

Le tableau 1 ci-dessous résume pour chacun des 4 réseaux que nous avons étudiés le nombre de réactions et de métabolites qui les composent et le nombre d'*efms* que nous avons trouvés.

TABLE 1 – Nombre de réactions, métabolites (total et internes) pour les réseaux de la mitochondrie : muscle, foie, levure, et pour le réseau du métabolisme central de la plante.

Noms	Nb. Réactions	Nb. Tot Métabo.	Nb. Métabo Int.	Nb. EFMS
Mito. Muscle	37	52	31	3 253
Mito. Foie	44	61	36	2 307
Mito. Levure	40	59	34	4 637
Plante	78	70	55	114 614

Les calculs des *efms* ont été obtenus grâce au logiciel regEfmtool [1]. Cette nouvelle version du logiciel Efmtool [2] [7] permet de calculer très efficacement de très grand réseau, éventuellement en utilisant des règles logiques de contraintes. Si historiquement ces calculs étaient réalisés avec l'aide du logiciel metatool puis de sa nouvelle version CellNetAnalyser, les limitations dûes à l'implémentation MatLab des algorithmes rendent ce logiciel très peu utilisable pour les réseaux de grandes tailles. Jungreuthmayer et al [8] ont montré l'intérêt de l'implémentation de regEfmtools

1. téléchargeable à partir de la page `http://www.biotec.boku.ac.at/regulatoryelementaryfluxmode.html`
2. téléchargeable à partir de la page `http://www.csb.ethz.ch/tools/efmtool/`

TABLE 2 – Nombre de réactions, métabolites (total et internes) pour les réseaux de la mitochondrie : muscle, foie, levure, et pour le réseau du métabolisme central de la plante.

Noms	Nb. EFMs	Long. Moyenne	Long. Min	Long Max
Mito. Muscle	3 253	17	2	23
Mito. Foie	2 307	16	2	24
Mito. Levure	4 637		4	22
Plante	114 614	37	2	53

dont les temps de calcul sont de l'ordre de quelques dizaines de minutes quand l'implémentation MatLab requière plusieurs heures, quand les calculs se terminent, ce qui n'est pas toujours le cas.

Malgré tous les problèmes causés par la génération de ce grand nombre d'*efms*, nous tenons à souligner leur réel intérêt en rappelant que dans la thèse de Sabine Pérès [9], il a été montré que dans l'ensemble des efms des 3 réseaux modélisant le métabolisme mitochondrial, il existe plusieurs *efms* correspondant au mutant décrit par Swimmer et al. [10]. Ce mutant permet de produire de l'`ATP` grâce au cycle de Krebs (réaction `R12`) en l'absence d'`ATP synthase` (réaction `R3`). Trouver des *efms* correspondant à des voies *alternatives* prouve formellement que ces voies sont valides dans le réseau et donc peut conforter les résultats biologiques en éloignant le spectre du résultat obtenu par hasard ou erreur de mesure.

Traitement des résultats obtenus

Le calcul des modes élémentaires de flux d'un réseau métabolique donné fournit une nouvelle vision de ce graphe en permettant par exemple d'expliciter les *"shunts"* ou les solutions alternatives existants. De nombreux travaux tentent actuellement de rendre l'analyse plus aisée en découpant par exemple le réseau en modules plus petits [11]. Si cette solution rend parfois les résultats plus intelligibles, elle a l'inconvénient de ne pas être complète puisque bien évidemment les solutions inter-modules (qui ne sont pas obligatoirement la somme des solutions de chaque module) ne sont pas données. Il apparaît donc que la mise en oeuvre d'outils d'analyse automatique des ensembles d'efms obtenus est indispensable pour être réellement utilisable dans le cas des réseaux faisant intervenir plusieurs voies.

Analyse statistique

L'analyse de grandes masses de données est très généralement réalisée au moyen de statistiques descriptives qui permettent de mieux appréhender les résultats obtenus. Dans cette optique, nous avons réalisé pour chaque réseau métabolique étudié, un ensemble de traitement afin de caractériser les résultats obtenus lors du calcul des *efms*.

Calcul des longueurs moyennes Les *efms* étant des chemins minimaux, leur longueur est un bon indicateur de la somme des transformations nécessaires et suffisantes pour aller d'un métabolite *entrant* à un métabolite *sortant* car il n'y a pas à craindre de *bruit* causé par des redondances ou cycles. Nous pouvons observer non seulement une certaine variété entre les 3 exemples mitochondriaux mais surtout lorsqu'on analyse les résultats obtenus pour le réseau de la plante, que la longueur évolue avec la taille du réseau. Ce résultat n'est pas forcément évident car augmenter le réseau signifie en général ajouter des voies métaboliques, encore une fois souvent étudiées séparément, et non étendre chacune de ces voies. On peut expliquer cette augmentation de la taille des *efms* par le fait que l'on doit équilibrer les métabolites, y compris ceux souvent négligés comme le `CO2` ou l'`ATP`, et qu'en ajoutant des réactions on ajoute très souvent de nouvelles contraintes sur ces métabolites.

Calcul des occurences des réactions Pour mieux caractériser la structure d'un réseau, on peut examiner le taux de participation d'une réaction à l'ensemble de solutions obtenues par le calcul des efms. On peut alors s'intéresser aux réactions toujours (ou massivement) présentes qui pourraient être assimilées à des sortes de *hubs* dont l'activité serait des points de contrôle du réseau. Les réactions ne participant à aucun efm sont également intéressantes puisque cela signifie qu'aucun chemin valide dans le graphe ne peut les utiliser. Cela pose alors la question de la validité de la description du réseau. A cette occasion, nous soulignons que la mise au point de cette description : choix des métabolites internes ou externes, choix de la réversibilité ou non des réactions, est un point essentiel de la modélisation des réseaux métaboliques et que le calcul des efms est un outil extrêmement utile pour vérifier/valider cette modélisation. En effet, en détectant ainsi des réactions ne pouvant jamais participer à un chemin équilibré, ce calcul permet d'identifier des connexions dans le graphe qui ne sont pas valides. Il n'est pas possible d'envisager de découvrir ces problèmes simplement en *"regardant"* le réseau car le graphe est d'une taille trop importante pour cela.

Analyse des équations bilan. Il est possible d'obtenir à partir d'un *efm*, l'équation bilan qui lui correspond. Le terme équation bilan doit ici être pris au sens biochimique, c'est l'ensemble des métabolites externes en entrée, nécessaires à la réalisation de l'efm et l'ensemble de ceux qui sont produits. Nous avons analysé cette information car il est intéressant de noter que bien que chaque *efm* soit unique, cela conduit à des doublons dans l'ensemble des équations bilan [3] apportant ainsi une preuve irréfutable que des ensembles différents de réactions (formant des voies valides différentes) conduisent bien à des ensembles de métabolites d'entrée/sortie identiques. Ainsi, dans le cas de mesure de flux métaboliques, il est indispensable de prendre en compte que la seule mesure des métabolites externes se garantit pas l'identification des protéines qui ont été activées. C'est aussi la preuve que lorsque certaines protéines sont *non disponibles* pour effectuer une réaction, que ce soit pour des problèmes de conformation ou parce que l'ensemble des substrats nécessaires ne sont pas accessibles, il est tout à fait possible qu'une *"variation"* de la voie métabolique se mette en place de façon plus ou moins permanente. Pour les réseaux étudiés, en moyenne 4 à 5 efms exhibent la même équation bilan, avec bien sûr des efms qui restent uniques et un maximum du nombre d'efms ayant la même équation bilan pouvant aller jusqu'à 10. C'est cette observation qui nous a conduit à considérer les efms au travers des métabolites qu'ils utilisent.

Ensembles de réactions communs à différents efms Le calcul des efms permet d'identifier des groupes de réactions qui sont toujours associés dans un chemin valide (appelés *subsets* dans le logiciel metatool). Bien qu'en général limité à un petit nombre de réactions, cela permet tout de même d'obtenir quelques simplifications du réseau. Dans nos réseaux, nous avons trouvé pour le muscle, le foie, la levure et la plante, resp. 7, 8, 6, 12 *subsets* réduisant le nombre de réactions à resp. 26, 28, 26, 52. Si des réactions ne sont pas toujours associées dans un efm, elles peuvent l'être souvent, construisant ainsi des *motifs* de réactions communs à un groupe d'efms. L'identification de ces motifs fait l'objet de la section suivante.

Recherche des motifs dans les efms

Il existe un grand nombre de méthodes de classification qui permettent de construire des ensembles en fonction de critères de similitude. Des méthodes tel que le clustering hiérarchique sont couramment utilisées dans des domaines variés - on citera la génomique ou la phylogénie dans le domaine de la biologie.

Malheureusement, les caractéristiques même des modes élémentaires de flux : uniques et minimaux, en font des éléments difficiles à classer par les méthodes classiques. Par exemple si l'on

3. On notera que le logiciel Metatool a choisi de ne pas citer les métabolites qui sont à la fois en entrée et en sortie comme cela est généralement la norme en biochimie. Cette remarque est importante car deux bilans peuvent sembler identiques alors que ces métabolites équilibrés en entrée/sortie ne sont pas les mêmes. Il faut donc être vigilant sur ce point.

considère les méthodes de clustering classiques qui s'appuient généralement sur la construction d'ensembles disjoints, tenter de réaliser ce type de construction avec des *efms* se révèle quasiment impossible et le plus souvent fournit suivant notre expérience, un résultat de peu d'intérêt. En effet si l'on considère dans le graphe d'interactions, d'une part leur propriété d'être uniques et minimaux et d'autre part le fait que le nombre de solutions soit très grand relativement au nombre d'éléments, il est évident qu'un certain sous-ensemble de réactions est commun à différents *efms*. Un rapide test sur d'autres outils classiques comme la construction de treillis de gallois, se révèlent tout autant décevant, car l'explosion combinatoire du nombre de sous-ensembles interdit de tel calcul sur les ensembles d'*efms* de la taille de ceux que nous manipulons.

Toutefois désirant obtenir une classification des nos efms, nous avons conservé l'*idée* de trouver une méthode de type clustering qui soit utilisable. Utiliser de telles méthodes suppose la définition d'une métrique comme critère de ressemblance entre deux éléments. Le codage de la présence ou de l'absence d'une réaction dans un efm est codée par une valeur 0 ou 1 mais comme la réaction peut être utilisée de façon réversible dans l'efm, la valeur -1 est utilisée pour coder cette situation. Nous désirons un critère qui prenne en compte ce cas et aussi le fait que deux efms de longueur 3 ayant 2 réactions en commun, sont plus ressemblant que deux efms de longueur 2 ayant 1 une réaction en commun.

Nouvelle approche basée sur les coupes de graphes

Des travaux récents on ouvert une nouvelle voie dans l'analyse des voies métaboliques grâce à un calcul dual des modes élémentaires : le calcul des coupes minimales du graphe d'interactions. Cette thèse, porte en partie sur l'étude de cet outil.

Le calcul de *"Minimal Cut Sets"* ou MCSs, intègre la même hypothèse que les modes élémentaires de flux en ce qui concerne l'état stable du réseaux, mais au lieu de calculer les chemins possibles, il s'agit alors de calculer les ensembles minimaux de réactions qui déconnectent ce graphe. Il est possible de demander ce calcul pour une fonction objective ou sur l'ensemble du graphe.Le pari est que cet ensemble sera plus petit que celui des modes élémentaires, mais aussi que la taille des MCSs sera en moyenne plus petite que celle de EFMs et donc permettra une analyse plus aisée.

TABLE 3 – **Comparison of the number and the length of EFMs and MCSs**.

Network	**Nb. EFMs**	**Nb. MCSs**	**Nb. MCSs with** `Glc_up`
Vss	22,469	13,901	15
Vac_f	34,752	14,446	15
Vac_g	1,246	562	561
Vac_s	19,392	14,473	15
Vgl_out	19,608	5,500	87

Nous avons réalisé le calcul des MCSs sur nos différents réseaux. La table 3 montre que pour des réseaux dont le nombre de EFMs n'est pas gigantesque, de l'ordre de quelques milliers, nous n'observons malheureusement pas de diminution du nombre d'éléments à observer. Toutefois, dans le cas du réseau de la plante dont le nombre de EFMs dépasse la centaine de milliers, non seulement le nombre de MCSs est inférieur mais surtout la taille des MCSs ne semble pas croître avec la taille du réseau, ce qui nous semble être le résultat le plus intéressant de cette méthode. Malheureusement la recherche de motifs communs grâce à l'algorithme ACOM ne donne pas de résultat satisfaisant, ceci est très probablement dû à la petite taille des MCSs ne permettant pas

la même liberté sur les paramètres de cet algorithme et rendant son réglage très délicat. Nous avons réalisé des statistiques descriptives des MCSs obtenus. Ainsi il est toujours intéressant de répertorier les réactions qui n'appartiennent jamais à un MCS. Cela signifie que le réseau ne peut jamais être déconnecté au moyen de cette réaction. On peut donc en déduire que construire un mutant qui inhibirait ces réactions n'aurait pas d'effet sur le comportement général du réseau métabolique. Les réactions toujours présentes dans les MCSs sont par ailleurs indispensables au fonctionnement du réseau, mais ceci peut bien sûr être également observé dans les efms. Les couples ou les triplets de réactions (on ne considèrera pas les MCSs de taille 1 dont l'interprétation est triviale) sont intéressants à étudier car ils fournissent un résultat très facile à exploiter pour les biologistes. Un couple ou un triplet de réactions qui constituent un MCSs peut couper toutes les voies possibles dans un réseau, cette information permet de mieux comprendre l'activité de ce réseau surtout si ces réactions ne sont pas directement reliés aux mêmes métabolites. Pour mieux expliquer ceci voici un exemple très simple du TCA cycle (ou cycle de Krebs).

Étude de cas : production de sucres et acides aminées dans le fruit de tomate

A partir des résultats obtenus à la fois dans le calcul des EFMs et des MCSs sur le réseau donné en annexe, nous avons sélectionné les EFMs permettant la production de 6 différents substrats ayant un intérêt dans l'étude du métabolisme du fruit de tomate dans le cas où il n'y a pas d'entrée de `Glucose` (réaction Glc_up). Pour ce faire, nous avons sélectionné pour chaque cas, les EFMs contenant la réaction responsable de cette production. Ces substrats sont Glucose, Fructose, Sucrose, Glutamine, Starch et les réactions concernées sont respectivement : Vac_c, Vac_s, Vac_m, Vss, Vgl_out.

La table 3 montre pour chaque cas les effectifs de EFMs concernés. En ce qui concerne les MCSs, nous avons sélectionné les MCSs qui contiennent Glc_up (puisque celui-ci est bloqué) et la réaction ciblée. A partir du résultat des MCSs de taille 2, nous avons identifié 8 réactions qui participent toujours à la production des 5 metabolites d'intérêt en absence d'entrée de glucose. Ces réactions peuvent être considérées comme le coeur du réseau. Ces réactions sont : **Vpgi**, **Vfbp**, **Vpgi_p**, **Vrbco**, **Tg6p**, **Vald**, **Vriso_p** et **Vepi_p**. En analysent les MCSs de taille 3, ajoutent une liste de 5 réactions qui sont ensuite une des alternatives possible pour les différent chemins possibles. L'utilisation conjointes des EFMs et des MCSs nous permet donc d'identifier des réactions hubs dans ce réseau.

Pour terminer ce chapitre, nous voudrions souligner l'importance de la qualité du code des différents outils utilisés. Les versions les plus récentes des concepteurs de la méthode des modes élémentaires ont fait le choix de privilégier des versions utilisants un environnement Matlab, malheureusement peu adéquate pour supporter les calculs lourds. Non seulement cette bibliothèque n'est pas très rapide mais surtout malgré une documentation affirmant que dans sa version unix, la taille de la mémoire n´était limitée que par la mémoire disponible sur la machine, nous avons constaté qu'il n'en était rien. Les calculs sur le réseau de la plante sont quasiment impossible à obtenir avec les versions de CellNetAnalyser sous Matlab. Fort heureusement, il existe d'autres versions du calcul des EFMs, entre autre celle écrite en langage java par Marco Terzer [7] mais elle est peu documentée. Plus récemment, Christian Jungermeyer [8] a produit une bibliothèque de fonctions intégrant EFMtools et une extension qui permet d'écrire un ensemble de règles logiques pour calculer les EFMs avec des contraintes fonctionnelles. Dans le même environnement, mais cette fois écrit en langage C, on dispose aussi du calcul des MCSs et ce de façon très performantes. L'ensemble des calculs regEFMtools et mcsCalculator, font en général passer les calculs de plusieurs heures avec CellNetAnalyzer (quand ils terminent) à moins d'une minute.

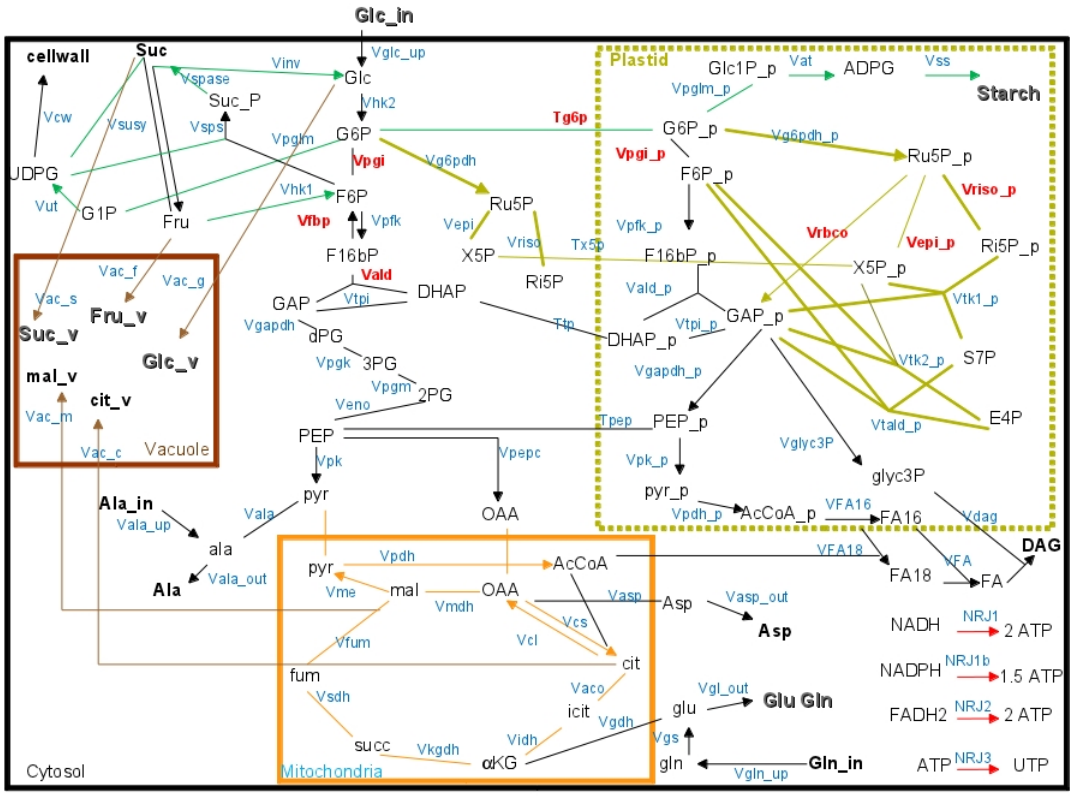


FIGURE 2 – **Enlarged metabolic network of a heterotrophic plant cells with 8 mandatory reactions highlighted.**

Conclusion

L'analyse de la structure statique des réseaux permet de mieux identifier le niveau de complexité auquel se situe les réseaux métaboliques. En effet, le passage de l'étude d'une réaction à celle de la voie métabolique puis d'un ensemble de voies constituant un métabolisme ne génère pas une complexité qui croit linéairement mais bien exponentiellement bien que l'ajout de noeuds modifie peu les paramètres classiquement étudiés en théorie des graphes comme l'arité moyenne des noeuds ou le diamètre du graphe. Les outils comme la recherche de chemins minimaux dans le graphe, les EFMs, permettent d'identifier cette complexité mais les résultats obtenus restent encore difficile à analyser entre autres à cause de leur taille. La combinaison de l'analyse des EFMs et des MCSs permet d'identifier les réactions les plus essentiels pour produire un metabolite d'intérêt. Notre analyse du réseau du fruit de tomate a montré que malgré la taille des données à manipuler il était possible d'en extraire des informations utiles qui peuvent ensuite être prise en compte dans l'interprétation des expériences biologiques qui sont conduites.

Références

[1] A. Wagner and D. A. Fell, "The small world inside large metabolic networks," in *Proceedings of the Conference of The Royal Society in London B*, vol. 268, Apr. 2001, pp. 1803–1810.

[2] J. L. Peterson, "Petri Nets," *ACM Comput. Surv.*, vol. 9, no. 3, pp. 223–252, Sep. 1977. [Online]. Available : http://doi.acm.org/10.1145/356698.356702

[3] I. Koch and M. Heiner, "Petri Nets," in *Analysis of Biological Networks*. Wiley, 2008, ch. 7, pp. 139–179.

[4] E. Grafahrend-Belau, F. Schreiber, M. Heiner, A. Sackmann, B. H. Junker, S. Grunwald, A. Speer, K. Winder, and I. Koch, "Modularization of biochemical networks based on classification of Petri net t-invariants." *BMC Bioinformatics*, vol. 9, no. 1, p. 90, Jan. 2008. [Online]. Available : http://www.biomedcentral.com/1471-2105/9/90

[5] J. Stelling, U. Sauer, Z. Szallasi, F. J. Doyle 3rd, and J. Doyle, "Robustness of cellular functions." *Cell*, vol. 118, no. 6, pp. 675–685, 2004.

[6] T. Wilhelm, J. Behre, and S. Schuster, "Analysis of structural robustness of metabolic networks," *Systems biology*, vol. 1, no. 1, pp. 114–120, 2004.

[7] M. Terzer and J. Stelling, "Large-scale computation of elementary flux modes with bit pattern trees," *Bioinformatics*, vol. 24, no. 19, pp. 2229–2235, 2008. [Online]. Available : http://bioinformatics.oxfordjournals.org/content/24/19/2229.abstract

[8] C. Jungreuthmayer, D. E. Ruckerbauer, and J. Zanghellini, "regEfmtool : Speeding up elementary flux mode calculation using transcriptional regulatory rules in the form of three-state logic," *Biosystems*, vol. 113, no. 1, pp. 37–39, 2013. [Online]. Available : http://www.sciencedirect.com/science/article/pii/S0303264713000890

[9] S. Pérès, "Analyse de la structure des réseaux métaboliques : application au métabolisme énergétique mitochondrial," Ph.D. dissertation, Université de Bordeaux 2, 2005.

[10] C. Schwimmer, L. Lefebvre-Legendre, M. Rak, A. Devin, P. P. Slonimski, J. P. Di Rago, and M. Rigoulet, "Increasing mitochondrial substrate-level phosphorylation can rescue respiratory growth of an ATP synthase-deficient yeast," *Journal of Biological Chemistry*, vol. 280, no. 35, pp. 30 751–30 759, 2005.

[11] J. Gagneur and S. Klamt, "Computation of elementary modes : a unifying framework and the new binary approach." *BMC bioinformatics*, vol. 5, p. 175, 2004.

Contents

Introduction

Motivation

From the last ten years, research in biology has been characterised by the extraction of large amount of data concerning living processes. New machines more and more powerful allow whole genome sequencing, isotopic tagging of large pools of molecules and building the trace of applying transformations, measuring of the transcriptomic activities and so on. A consequence to this evolution is that now it is not at all possible to manage by hand this amount of data. Bioinformatics and more precisely System Biology offer methods to create automatic analysis of processes and new tools to visualise the obtained results, in large quantity too. In this context, our team focuses on the modelling of biological networks and more specifically metabolic networks, this problem is the main purpose of this PhD thesis manuscript.

In metabolic networks, the different interactions between molecules create networks which even if they are not so big, comparing with social networks or communication ones, can be qualified as complex. This is done because the high number of connections between the elements belonging to the network lead to many difficulties to measure consequences when one of these elements is disrupted (for example by a genetic mutation). Different methods exist to design metabolic networks and their behaviours. The most traditional one consists on building a set of algebraic or differential equations which describe the evolution of molecules concentration during the time course. These systems are suitable to model a small number of reactions but not to build a model at the level of the whole cellular organism (around one hundred of reactions). Tools like elementary flux modes proposed by Schuster, allow to identify sets of biochemical reactions (parts of the network) which satisfy specific biological constraints (network steady state in the case of metabolism). A metabolic function pathway could be model by one or several elementary flux modes. Properties of elementary modes: uniqueness and minimality, give to them the capacity to be performed tools to analyse the network structure. From elementary modes it is possible to show the robustness of some biological pathways or the major role of some reaction *hubs*. The main problem with this method is the time to compute the elementary modes, the

method is based on linear algebra and the computation time can be exponential (depend on the number of reactions). Moreover, we also need a large amount of memory as the results is often several hundreds of thousands of solutions.

For this work, the first investigation that has been done in metabolic pathway analysis is to compute topological properties of experimental networks. We have been also interested to study elementary flux modes method proposed by Schuster and applied it in finding feasible pathways inside some metabolic networks: TCA cycle, mitochondria networks and heterotrophic plant cell metabolism. This method seems fitting in the analysing of feasible pathways in small networks, e.g. TCA cycle, mitochondria tissues networks, but the number of elementary flux modes is so huge that we cannot handle them manually in the case of metabolic network of heterotrophic plant cell. Hence, it requires another method to treat these elementary flux modes. We have chosen minimal cut sets approach proposed by Klamt. Fortunately, the number of minimal cut sets trends decreasing when the network size grows up. The bottom line of minimal cut sets is to give the set of reactions which size is smaller than those of elementary flux modes. Therefore, the collaboration of EFMs analysis and MCSs computation could be considered as the main protocol that we have used in this research.

Organisation of the thesis

This thesis is divided into four main chapters which are the PhD research results. A brief description follows of what is to be found in each chapter. The contents of individual chapters are summarised as follows.

Chapter 1 gives the principles of context of systems biology, metabolism, metabolic networks and basic concepts need to go through this research. This chapter also overviews of some traditional models of biological networks. It is ended by introducing commonly used approaches of metabolic network analysis.

Chapter 2 displays an overview of graph and complex networks. Subsequently, it gives some results of computing and comparing global structural properties, network centralities and investigating complex network features of mitochondria and metabolic networks.

Chapter 3 presents Minimal Cut Sets concepts and the algorithms to compute Minimal Cut Sets inside metabolic networks. Afterwards, the results computed on given metabolic networks are debated. The idea behind the collaboration of elementary flux modes and minimal cut sets is also discussed at the end of this chapter.

Chapter 4 presents the main application of this work by employing the structural analyses stated in Chapter 2 on heterotrophic plant cell metabolism and using of minimal cut sets

method stated in Chapter 3 in studying elementary flux modes. It shows the obtained results to characterise the architecture of the plant cell network that we have studied taking into account some specific productions of sugar.

Chapter 1

Metabolic Networks and Their Specifications

Living organisms are distinguished by their specified complexity.

Leslie Orgel
— The Origins of Life

Metabolism is one of the most important cellular processes. Informally speaking, metabolism is composed of many coupled and interconnecting biochemical reactions. A metabolic network is constituted by linked series of reactions that use up small molecules, the *metabolites*, and convert them into some another ones in a carefully defined fashion [19]. These reactions and metabolites belong to *metabolic pathways* which the large number of connections between reactions via substrate and product metabolites makes metabolic networks complex to study manually.

1.1 Context

Biology is a natural science concerning in the study of living things and their vital processes. It appears as a combination of observations and experiments. Its purpose is not only to inspect elements which compose living organisms but to investigate the relations between these elements. Nowadays, scientists have been facing extremely the explosion of information which is encountered in most applied sciences.

➤ **Systems Biology** is an engineering approach applied to biomedical and biological scientific research by using the computational and mathematical modelling of complex biological systems. The main studied objects of systems biology are living cells, which are viewed as integrated and interacting networks of genes, proteins, metabolites and biochemical reactions [120]. Integrating the investigations of these individual components or aspects of the system can enable to gain a deep insight on living organisms at multiple regulatory levels instead of studying individual aspects. In particular, people are interested in understanding how these complex interactions give rise to the function and behaviour of living cells. Genomics provides an overview of the

complete set of genetic instructions provided by the DNA, while transcriptomics looks into gene expression patterns. Proteomics studies dynamic protein products and their interactions, while metabolomics is a step in understanding organism's entire functioning. Figure 1.1 shows the depending on the level of studying which biological object/function is addressed from DNA to metabolites production.

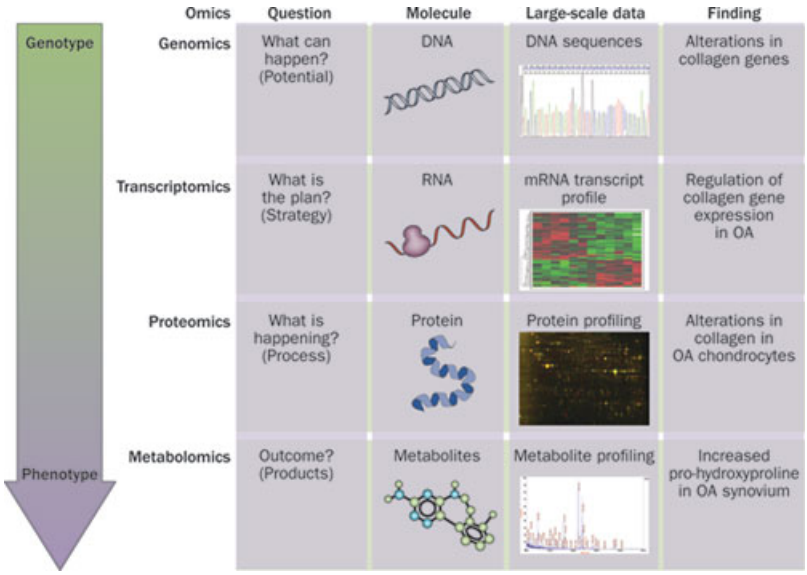


Figure 1.1: Levels of studying in systems biology. As genomic, transcriptomic, proteomic, and metabolic methods become more widely used, a critical need arises to integrate and analyse diverse data from multiple experimental sources using interdisciplinary tools.

Currently, researches on metabolic networks are interesting both for experimenters like biologists and for bioinformaticans. In bioinformatics, analysis of biological networks can be considered relevant to graph theory and a lot of methods have been used to model such networks. Before to present some of them, we begin to give some biological views of metabolic processes.

1.2 Metabolism

Metabolism is the set of life-sustaining chemical transformations within the cells of living organisms. It is often divided into two broad categories: *catabolism* and *anabolism* [19]. Catabolism is the degradation pathways to salvage components and energy from biomolecules such as nucleotides, proteins, lipids and polysaccharides, the process generates energy. Anabolism is the biosynthesis of biomolecules such as nucleotides, proteins, lipids and polysaccharides from simple precursor molecules, this process requires energy. Catabolism and anabolism are working together in cellular metabolism. A *metabolic reaction* is a chemical transformation occurring in living organisms, allowing them to feed, grow and regenerate. Metabolic reactions regulate

nearly all metabolic activities and are responsible for the building of complex molecules, for the breakdown of large molecules into smaller ones and for the yield of energy as well [94].

1.2.1 Molecules

Metabolites are small molecules which are implied in metabolic reactions. They are called *substrates* if they are used up by the reactions and *products* when they are the results of the transformation process. The *stoichiometric coefficient* of a metabolite in a reaction is the amount of that substance occurring in terms of molecule. The *flux* of a metabolic reaction is the rate of consumption of any substrate divided by the corresponding stoichiometric coefficient. This is equal to the rate of formation of any product divided by the corresponding stoichiometric coefficient. A reaction with a high flux operates at a faster speed than a reaction with a low flux. In addition, a flux is positive (resp. negative) if the forward (resp. backward) reaction is faster than the backward (resp. forward) reaction. While fluxes through reversible reactions may be negative, the convention is to consider that fluxes through irreversible ones are always non-negative [108].
Enzymes are protein complexes that serve as biological catalysts, that is, they speed up chemical reactions without undergoing any net chemical change during the reaction [94]. Without enzymes, most metabolic reactions would simply proceed too slowly at normal body temperature to support life.

1.2.2 Processes

➤ **Enzymatic reactions** Biological catalysts act by attaching the reaction molecules. Enzymes are often specific, meaning that each enzyme catalyses a single reaction or a very limited class of reactions. The specific three-dimensional shape of an enzyme is such that only the substrates it acts upon can fit into its active site - the particular portion of the enzyme that binds the substrates. After catalysing the reaction, the enzyme releases the products of the reaction. The enzyme remains intact in the process and can immediately bind fresh substrates. Thus, an enzyme molecule can be used over and over again. Enzymes increase the rate of chemical reactions by lowering the energy needed to activate the reaction. Enzyme activity is influenced by a large number of factors. Environmental conditions, such as pH, temperature, or salt concentration may change the three dimensional shape of an enzyme, altering its rate of activity and/or its ability to bind substrate.

➤ **Carrier and Channel** Most of reactions are within cells and often occurred inside a specific *compartment*, for instance *cytosol* or *mitochondrion*. Transporters serve to interconnect them and ensure availability of certain metabolites among compartments. The energy exchanged inside a cell between two cellular compartments separated by a membrane are not enzymes reactions. It exists embedded proteins which allow molecules to transverse the membrane.

We distinguish two categories of these proteins to ensure this movement: *carrier proteins* and *channel proteins*.

✓ A *carrier protein* is the protein that transports specific substance through intracellular compartments, into the extracellular fluid, or across the cell membrane. Carrier proteins are involved in facilitated diffusion and active transport of substances out of or into the cell (e.g. diffusion of sugars, amino acids and nucleosides, uptake of glucose, transportation of salts, glucose, amino acids, etc.).

✓ A *channel protein*, also called *transporter*, is the protein responsible for mediating the passive transport of molecules from one side of the lipid bilayer to the other. In some cases, molecules pass through channel proteins that span the membrane.

1.3 Basic concepts in Metabolic Pathways Analysis

In order to follow easily computational models and analysing approaches of metabolic networks, it is necessary to preview some relevant concepts.

1.3.1 Metabolic Pathways and Networks

A metabolic pathway is a series of connected enzymatic reactions that produce one or several specific products. Metabolic pathways are often referred biological functions which are widely shared by all organisms even though variations in the list of reactions are observed depending on the species. For example, the well-known *glycolysis* pathway is a catabolic process, it is the transformation of *glucose* molecules into *pyruvate*. Figure 1.2 shows a version of the *Arabidopsis Thaliana glycolysis* pathway where the metabolites are mentioned with their names and the reactions with numbers (the EC number[1]). The conversion of *glucose* substrate to *pyruvate* requires in this model 10 main reactions but a lot of other reactions are needed to produce some co-factors metabolites required in different reactions. The connections with another pathways are mentioned as boxes with the pathway name inside.

At this time, the Kyoto Encyclopedia of Genes and Genomes (KEGG) database[2] contains more than $9,700$ biochemical reactions, $2,900$ reaction classes and $6,000$ enzymes through several thousands of pathways. It shows the enormous amount of information that has to be taken into account. Moreover, even several databases like KEGG or MetaCyc have stored the pathway descriptions in electronic formats, most of available descriptions are represented in paper-based textual forms found in biochemistry textbooks [124, 188] or in static images that make them difficult to use and handle.

[1] The Enzyme Commission assigned each enzyme a recommended name and a 4-part number depending on their activity. 6 main groups classify the main enzyme functions, the sub-numbering refers the location, then category... see (http://www.chem.qmul.ac.uk/iubmb/enzyme/) for a full explanation.

[2] http://www.kegg.jp

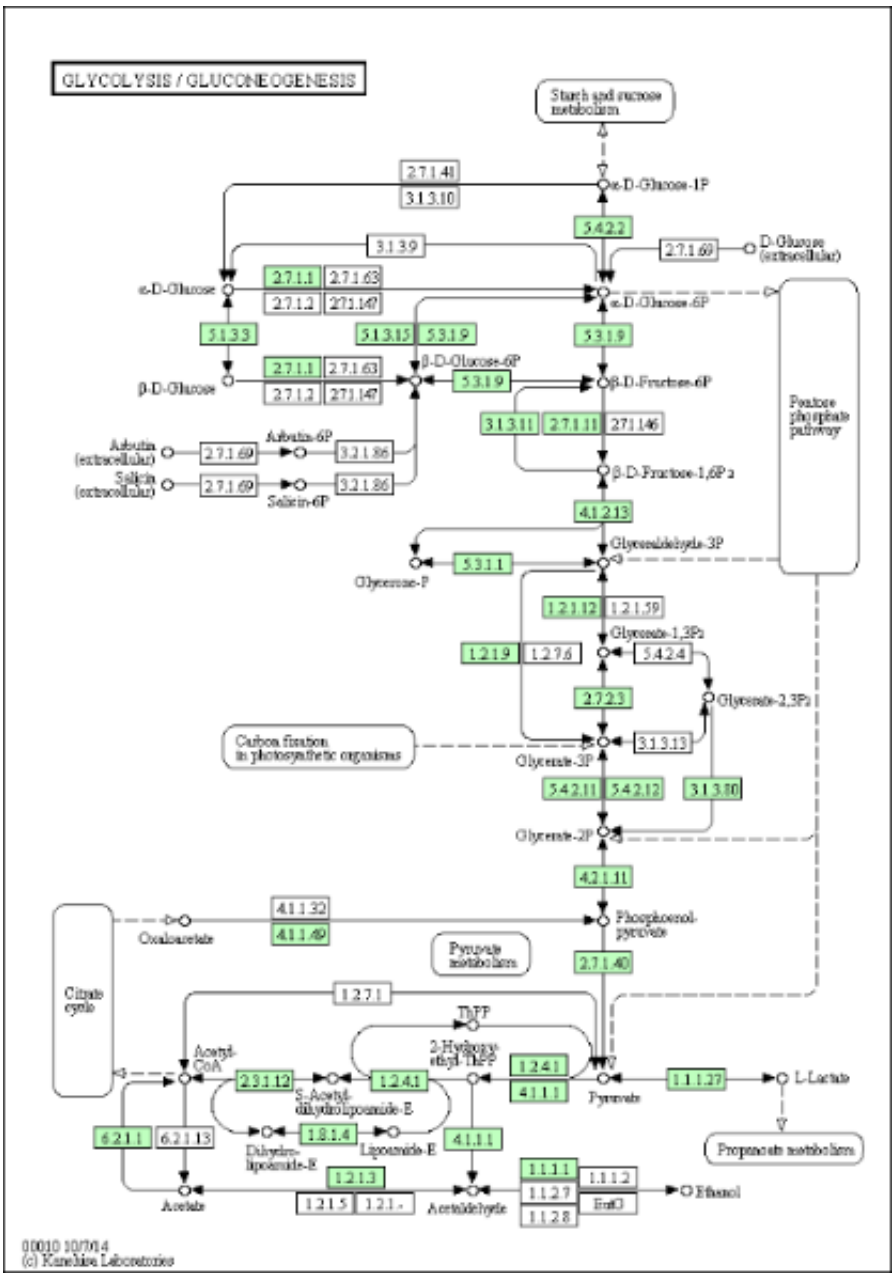


Figure 1.2: A model of Arabidopsis Thaliana glycolysis from KEGG website

➤ **A metabolic network** is build of a set of metabolic pathways. Nowadays, it is possible to reconstruct the network of biochemical reactions in various organisms from bacteria to human beings by sequencing of complete genomes. A number of such networks are available in online biological databases such as KEGG [90], EcoCyc [98], BioCyc [95], metaTIGER [180], etc. Such metabolic networks are useful tools for studying and modelling metabolism. Their applications have been employed in finding drug targets in cancer using profiling metabolic networks [52, 75], in metabolic engineering [173]. In this work, we have studied several cases of metabolic networks like the models of mitochondria metabolism depending on some tissues of human being and

yeast, or the central metabolism of heterotrophic plant cell. These different networks exhibit different sizes and levels of complexity. We have chosen them to test network analysis tools and evaluate their capability of use in large-scale networks. The next section presents a way to encode metabolic networks in the aim to apply mathematical or computerised methods to characterise them.

Metabolic network as graph A metabolic network can be modelled as an undirected graph where metabolites (circles in Figure 1.2) are nodes and reactions are mentioned as edges (rectangles in Figure 1.2). This coding will be discussed more details in Chapter 2.

1.3.2 Metabolic networks features

From *in vivo* data collected, how to study and predict biological behaviours of living cells is often one of the primary challenges in systems biology. It requires mathematical models to translate these data to computational representations. The next paragraphs present some basic features used in this thesis.

➤ **Stoichiometric matrix** Metabolic networks composed of m metabolites and r reactions are usually represented by a stoichiometric matrix S of m rows and r columns.

Definition 1.1 (Stoichiometric matrix)**.** *Given a metabolic network, let m be the number of internal metabolites and let r be the number of all reactions in the network. The corresponding stoichiometric matrix is a matrix $S = (s_{i,j})_{1\leq i\leq m, 1\leq j\leq r}$, such that for each internal metabolite $i \in [1; m]$ and each reaction $j \in [1; r]$*

$$s_{ij} = \begin{cases} a & \text{if the reaction } j \text{ produces } a \text{ molecules of the metabolite } i \text{ and } a \in \mathcal{Q}^+ \\ -a & \text{if the reaction } j \text{ consumes } a \text{ molecules of the metabolite } i \text{ and } a \in \mathcal{Q}^+ \\ 0 & \text{otherwise} \end{cases}$$

$$S = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,n} \end{pmatrix}$$

Example 1.1. In [99], Klamt gives an example of how to write the list of reactions and its stoichiometric matrix from a given network. Inspired by this example, we consider a system of reaction equations declared in Table 1.1 and visualised in Figure 1.3. It consists of 8 reactions $(R1, R2, \ldots, R7, RSynth)$ and 5 internal metabolites A, B, C, D, E, while $S1$, $S2$, X, P represent external metabolites. The stoichiometric matrix of the system of these equations can be built as follows:

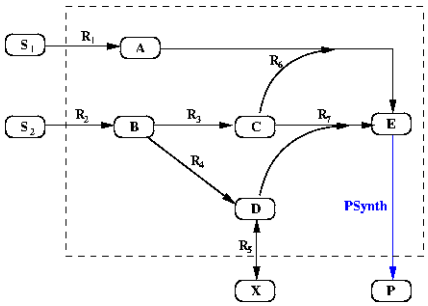


Figure 1.3: Network layout for a simple example of metabolic network (*NetEx*) designed by Klamt [99].

Table 1.1: An introductory metabolic network

Reactions
$R1$: $S1 \longrightarrow$ A
$R2$: $S2 \longrightarrow$ 2B
$R3$: B $\longrightarrow$ C
$R4$: B $\longleftrightarrow$ D
$R5$: D $\longleftrightarrow$ X
$R6$: A + C $\longrightarrow$ E
$R7$: C + 3D $\longrightarrow$ E
$RSynth$: E $\longrightarrow$ P

$$S = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{cccccccc} R1 & R2 & R3 & R4 & R5 & R6 & R7 & PSynth \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 2 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -3 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 \end{pmatrix} \end{array}$$

It should be noted that only internal metabolites are included in stoichiometric matrix.

➤ **Incidence matrix versus Stoichiometric matrix** Indeed, to represent metabolic networks some authors use incidence matrix [148] which coincides with the stoichiometric matrix [104]. As there are two types of metabolites: *internal* and *external*, an incidence matrix is built from the coefficients of the internal and external metabolites. Since the symmetry via the principal diagonal, we only take into accounts the coefficients to be above the principal diagonal. In other words, external metabolites are trivial in flux analysis because they do not operate at steady state.

➤ **Reaction Reversibility** The reversibility of a reaction is defined by the thermodynamic constraint. A reaction i is *irreversible* if and only if its flux is always non-negative, i.e. $r_i \geq 0$

[108, 170].

➤ **Dynamic mass balance** Mathematically, the temporal behaviour of a metabolic network can be described as a system of ordinary differential equations (ODEs). A compact expression of the system of these equations is defined as:

$$\frac{dX(t)}{dt} = Sv(X(t)) \tag{1.3.1}$$

where X denotes the m-dimensional vector of biochemical reactants and $v(X(t))$ is a r-dimensional vector of reaction rates which consists of nonlinear (often unknown) functions. These functions depend on substrate concentrations of metabolites.

➤ **Quasi-steady state** At the quasi-steady state (shortly called steady state) the mass balance in the network can be represented by the flux balance equation:

$$\frac{dX(t)}{dt} = Sv(X(t)) = 0 \tag{1.3.2}$$

where S is the stoichiometric matrix and v is the reaction fluxes.

1.4 Computational Models of Metabolism

The computational tools allow us gaining an in-depth insight into experimental results of molecular mechanisms of a particular organism. Actually, there have been lot of methods for modelling metabolic networks developed in the past decade (see more in [140]). We brief here the most well-known models used to represent metabolic networks. Figure 1.4 shows a way to group the modelling methods based on their scale and complexity accuracy rate.

1.4.1 A classical model: Kinetic Modelling

Probably the most straightforward and well-known model to metabolic networks modelling is to represent metabolic processes in terms of ordinary differential equations (ODEs). As we have seen in the previous section, the considered systems biology can be declared by a list of reaction equations, a description of reversible and irreversible reactions and a list of internal and external metabolites. This model can be considered as a bridge between structural modelling, which is based on the stoichiometry alone, and explicit kinetic models of cellular metabolism [147, 163].

1.4.2 Constraint-Based Models

The idea of constraint-based modelling is to describe a biological system by a set of constraints, which characterise its possible behaviours, but in general do not allow to make a precise prediction [25, 96, 125, 130]. Constraint-based modelling uses physiochemical constraints such as mass and energy balance, or flux limitations to describe the potential behaviours of an organism.

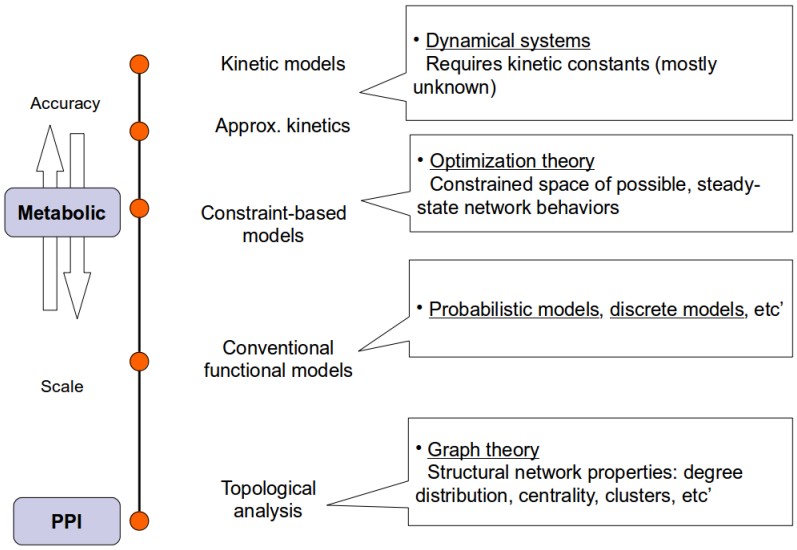


Figure 1.4: These approaches are grouped based on the accuracy and scale. [From a Lecture Notes of Tomer Shlomi, School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel, 2008.]

The classical starting point of constraint-based modelling is flux balance analysis of metabolic networks at steady state. Mathematically, this involves in computing a basis of the underlying polyhedral cone of the matrix. Existing methods focus on pointed cones, and often metabolic networks have to be reconfigured in order to obtain this property.

Indeed, constraint-based modelling has mainly focused on metabolism, and more integrative modelling approaches have been explored. Integrating this model with other approaches of modelling metabolism can be expanded the scope of quantitative prediction [1].

1.4.3 Conventional Functional Models

This modelling group is often included in probabilistic model, discrete models, etc. The most typical conventional functional approach is Boolean network modelling using to represent Gene Regulatory Networks (GRN) [37].

1.4.4 Graph-Based Models

To overcome the inherent limitations that happen in the construction of large-scale kinetic models, topological and graph-based approaches have remarkably interested recently [7, 17, 50, 82, 177]. Indeed, topological structure analysis of networks has a number of considerable advantages, as compared to the construction of explicit kinetic models. Topological network analysis does not presuppose any knowledge of kinetic parameters, thus it allows for an analysis of less well characterised organisms. It is applicable to extensively large systems, consisting of

several thousands of nodes, far beyond the realm of current kinetic models[3].

1.5 Approaches of Metabolic Networks Analysis

Studying metabolic networks is one of the leading tool in metabolic engineering. It supports to gain a comprehensive understanding of the control mechanisms of complex cellular metabolisms. Some approaches of metabolic networks analysis are addressed in this section.

1.5.1 Stoichiometric Analysis

A considerable improvement over purely graph-based models is the analysis of metabolic networks in terms of their stoichiometric matrix. In stoichiometric analysis [32, 33], one concerns the effects of the network structure on the behaviours and capabilities of metabolism. Questions that can be tackled include discovery of pathways that carry a distinct biological function from the network, discovery of dead ends and futile cycles, dependent subsets of enzymes. This approach allows us making identification of optimal and suboptimal operating conditions for an organism. Otherwise, it helps to analyse the network flexibility and robustness, e.g. under gene knock outs. The two most well-known variants of stoichiometric analysis are Flux Balance Analysis (FBA) and Elementary Flux Modes (EFMs) analysis that shall be addressed in the sections afterwards.

1.5.2 Flux Balance Analysis

Flux Balance Analysis (FBA) is a technique for analysing the flow of metabolites through a metabolic network [125, 128]. This method, which has been employed in a number of applications [127, 129], is based on constraints model. FBA calculates the flux of metabolites through the metabolic network, thereby getting enable to predict the growth rate of a given modelled organism or the production rate of a certain target metabolite. We show here briefly the principal idea behind the method in Figure 1.5.

In Figure 1.5, first of all, a metabolic network reconstruction is built (the figure **a**), consisting of a list of stoichiometrically balanced biochemical reactions. Next, this reconstruction is converted into a mathematical model by forming a matrix labelled S (the figure **b**). At steady state, the flux through each reaction is given by the equation $Sv = 0$ (the figure **c**). Since there are more reactions than metabolites in large models, there is more than one possible solution to this equation. In the figure **d**, an objective function is defined as $Z = c^T v$, where c is a vector of weights (indicating how much each reaction contributes to the objective function). In practice, when only one reaction is desired for maximisation or minimisation, c is a vector of zeros with a

[3]Schilling et al. [153] have developed a concept of *Extreme Pathways* actually closed to EFMs. The main difference between them is that *Extreme Pathways* has to double reversible reactions. That is why we have focus on only EFMs

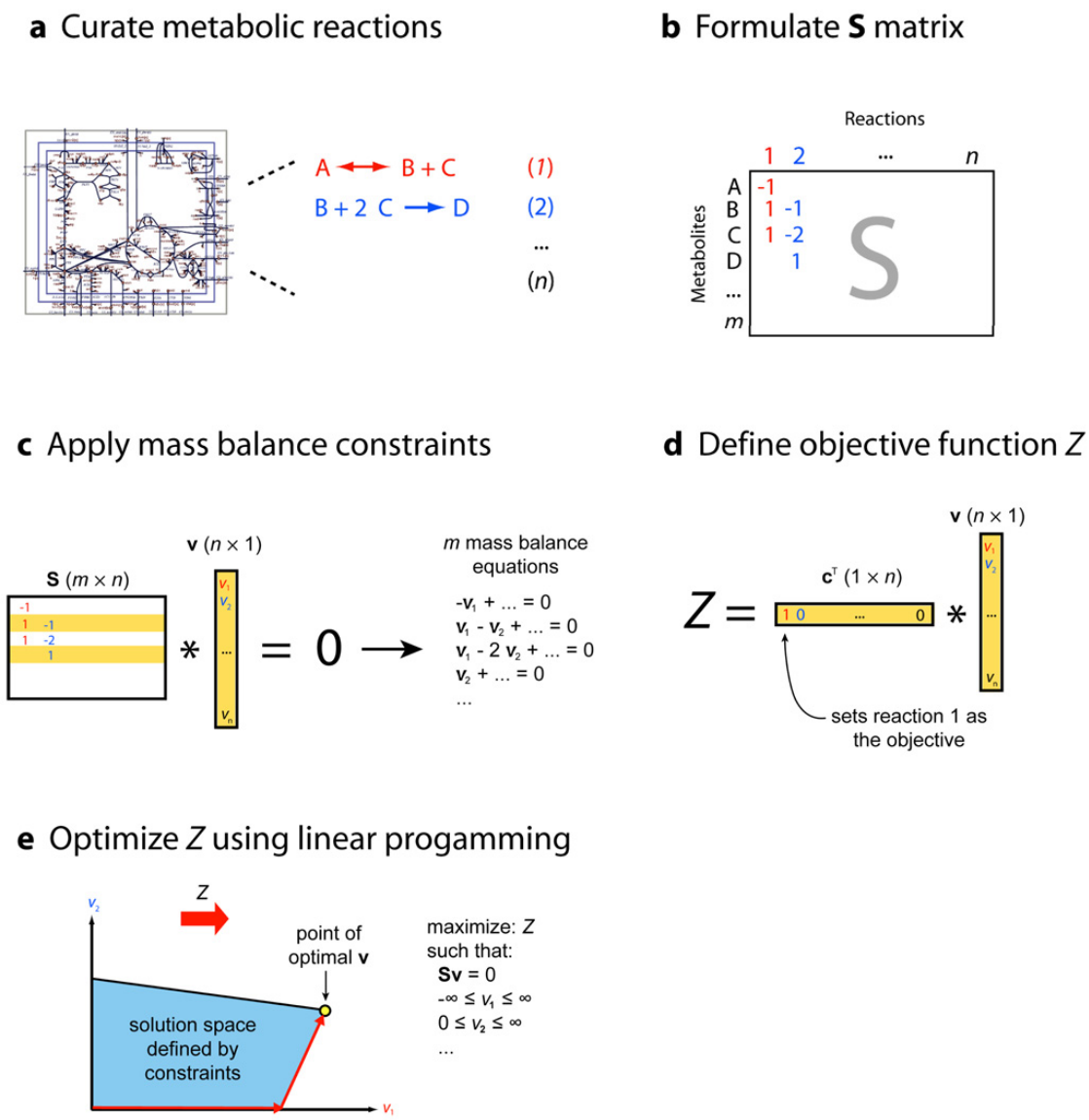


Figure 1.5: Formulation of an FBA problem. The image courtesy of Orth et al. [125]

one at the position of the reaction of interest. When simulating growth, the objective function will have a 1 at the position of the biomass reaction. Finally, linear programming can be used to identify a particular flux distribution that maximises or minimises this objective function while observing the constraints imposed by the mass balance equations and reaction bounds (the figure **e**).

1.5.3 Petri net

Petri net theory is a graphical and mathematical formalism suitable for the representation and analysis of dynamic networks at different abstraction levels [104]. It is used in various biology-related applications from analysing the dynamics of signalling pathways in cellular signalling networks [70, 150], genetic networks [30] to simulate metabolic pathways behaviours [104, 105].

The structure of Petri nets can be expressed linearly by a two dimensional matrix C (cf. Appendix C.2), of size $m \times n$. From this one, it is possible to determine structural properties of Petri nets like invariants. In a biological context, minimal *p-invariants* (place invariants) are used to model a kind of substrate conservation, while *t-invariants* (transition invariants) the concept of elementary flux modes (introduced in the next section).

The *t-invariants* describe the system behaviour of the network, e.g. for metabolic networks in the steady state. A *t-invariant* is defined as a vector $x \in \mathbb{N}^m$ which satisfies the equation $C.x = 0$. A *t-invariant* characterises a repetitive component of a model which is a set of transitions causing a return to a previous state of a model.

Similarly, a *p-invariant* is defined as a vector $y \in \mathbb{N}^n$ satisfying the equation $y.C = 0$. A *p-invariant* characterises a conservation component of the model. A conversation component is a set of places over which the weighted sum of the tokens is constant for every reachable marking. And *p-invariants* address conversation relations of metabolites in metabolic pathways models [150, 185]. In signalling pathway models, *p-invariants* can represent a different kind of conservation relation [150]. Enzymes occur inside biochemical processes in signalling pathways that makes a state changing to transmit a signal. The total concentration of all forms of an enzyme is modelled as a constant quantity considered as a marking invariant of a Petri net model. Thus, the *p-invariants* and their associated conservation components identify all the places representing a specific form of an enzyme [70].

The drawback of using ordinary graphs for representing biological networks, in general, and metabolic networks, in particular, is that they cannot capture the complex relationships between several nodes, for example multiple metabolites in a reaction or more than two protein interacting to form a complex. In addition, simple graphs do not provide an intuitive approach to study evolution of metabolic networks. An alternate is to use hypergraphs or bipartite graphs to represent metabolic networks [101].

1.5.4 Elementary Flux Modes Analysis

A related approach to FBA, Elementary Flux Modes (EFMs) analysis, was proposed by Schuster in 1994 [156] to analyse metabolic pathways. It is a constraint-based approach which can be used to calculate all biologically meaningful pathways through a network [155]. This method is useful to gain an insight into metabolism of living organisms and to identify all genetically independent pathways that are inherent in a metabolic network. By the definition, an EFM is a unique and non-decomposable set of reactions.

Let a metabolic network composed of r reactions and m metabolites and its stoichiometric matrx S. An unit $efm = (r_1, r_2, ..., r_k)$ is an elementary flux mode if it fulfils the following conditions [156]:

- Steady state: $S \times efm = 0$.
- Feasibility: For all i of an irreversible reactions, $r_i \geq 0$.
- Minimality: For all efm' of S, $supp(efm') \subseteq supp(efm) \Rightarrow \exists \alpha \in \mathcal{R}$ such that $efm' = \alpha \times efm$.

From the stoichiometric matrix, EFMs are computed by selecting groups of reactions which interact together and respecting the well-known steady-state mass balancing equation (cf. Equation (1.3.1)). Grafahrend-Beleau et al. [64] have shown that computing the set of EFMs of a given network is equivalent to compute the set of *t-invariants* of the network modelled through a Petri net.

In the small example network of TCA cycle given in Figure 1.6 (for details see [132]), we can see 15 reactions and 25 metabolites (11 internal and 14 external ones). Applying EFMs computation, 16 EFMs have been found. To analyse this result, for example we can consider the case of production of external `citrate` obtained by firing the transporter reaction $T1$. Figure 1.7 shows the 7 EFMs/available routes to fire $T1$ and if any of them can not be operated (by the inhibition of one reaction belonging to the EFMs) there is no more way to activate $T1$.

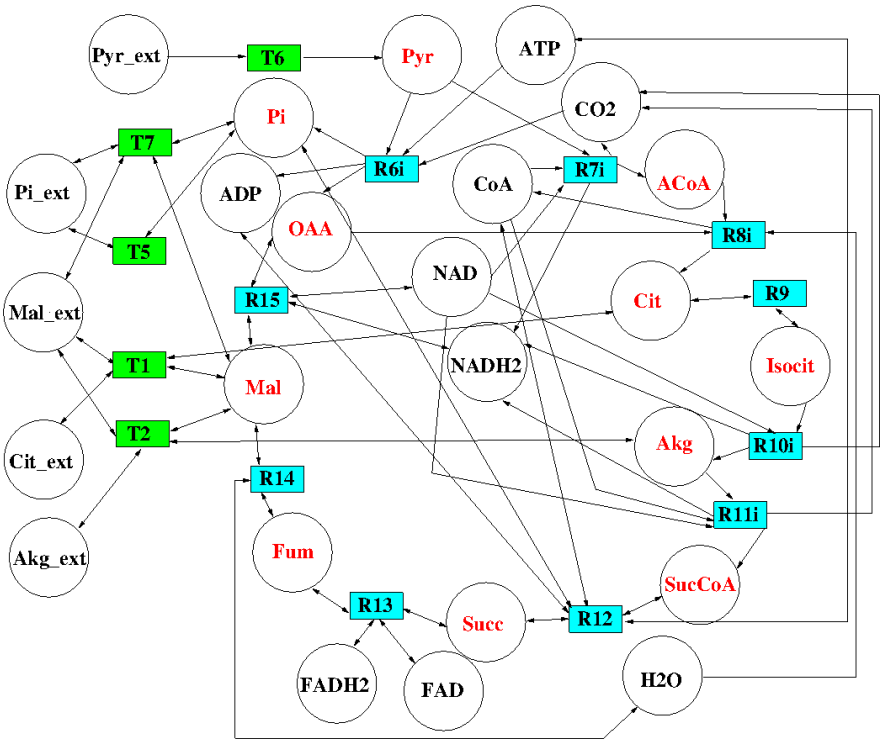


Figure 1.6: TCA cycle network. The metabolites are the circles: the red labels represent the internal metabolites, the other ones (black colours) represent the external metabolites. The rectangles are the reactions: the shapes filled by cyan are the transporters. The arrows denote the direction of reactions, a double arrow means that the reaction is reversible.

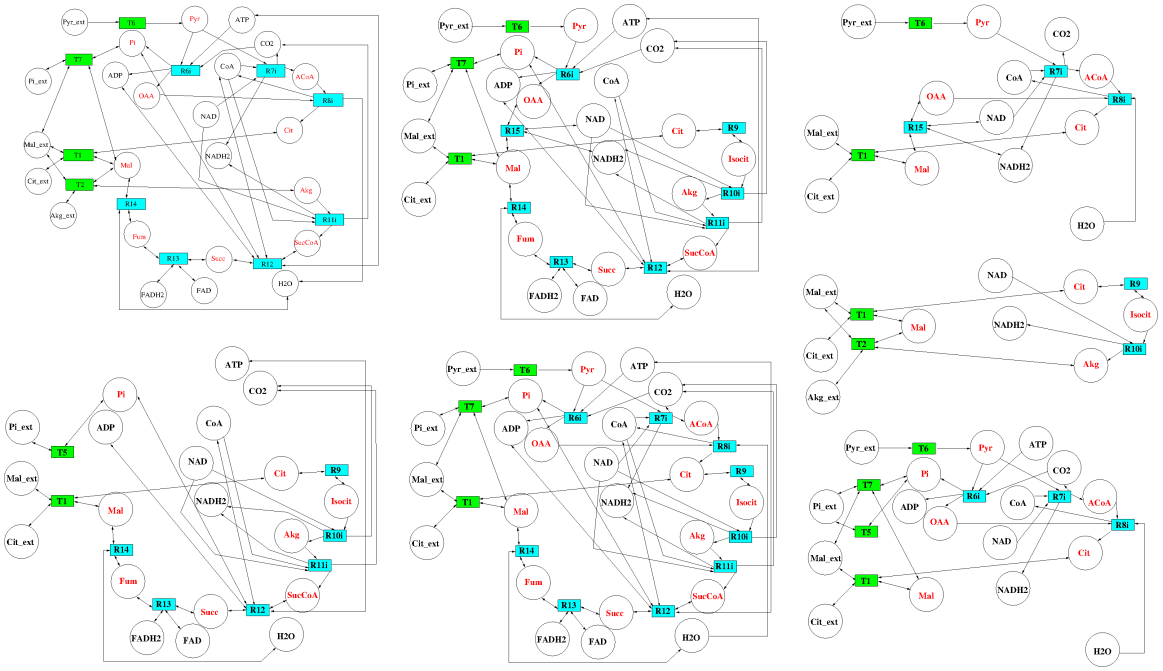


Figure 1.7: 7 EFMs containing T1 to produce external citrate.

A number of tools were specifically designed to compute all EFMs of a given metabolic network. They can be counted such as METATOOL [136], CellNetAnalyzer [102] (also known as a successor to METATOOL), and efmtool [167]. These tools implement an algorithm based on linear algebra and its complexity is exponential, especially for metabolic networks including many connected pathways [103]. As the number of obtained EFMs can be huge, enumerating all possible pathways that contain a given reaction is a difficult task [4]. The question of *"how to suit biological reasoning to such large results"* stays open. Classification of EFMs can be done by clustering methods. Due to their specificity: each EFMs is unique and minimal, classical hierarchical clustering does not offer satisfying results. Overlapping clustering seems to be more promising for this task. A classification method for EFMs, ACoM [135], has been proposed, based on motif findings with overlapping clustering tools. In [64] a classification of *t-invariants* is also studied using another agglomerative clustering algorithms. But most often the size of the results to classify still remains a major difficulty. Going back to classical graph theory methods, another way to extract knowledge from networks has been explored: computing graph diameter, average degree of nodes, average path length. . . and authors such as Barabási [17], Fell [50] or Jeong [82] have confirmed that metabolic networks can exhibit behaviours similar to small-world networks and can be explored that way to find organisation, links or hubs through metabolic networks [22]. More recent works have suggested using a dual view on the problem of finding feasible routes and of searching ways to cut access to a specific reaction and then to inhibit it. Chapter 3 details deeply this method that we have focused for this work.

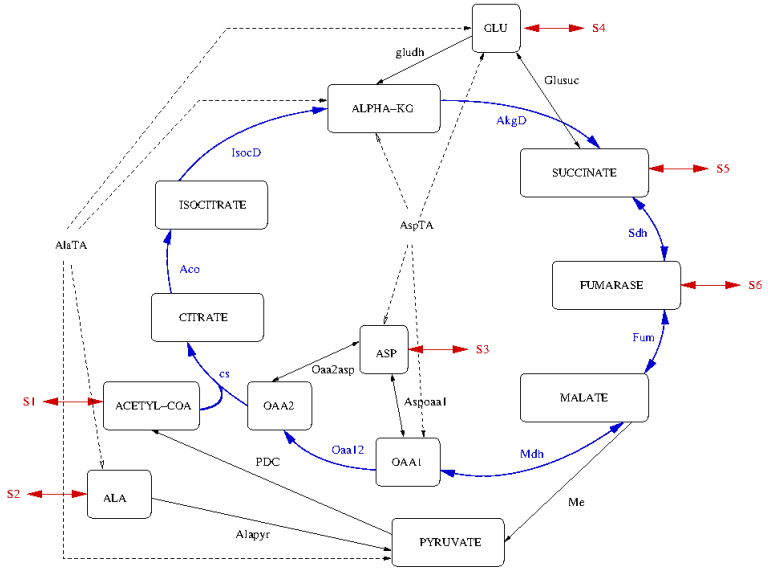


Figure 1.8: Metabolism of TCA cycle, designed by [183] and redrawn in [131]

1.6 Description of our experimental data

To evaluate the methods employed in our works, we have tested them on some datasets (i.e. different metabolic networks). All the networks that we have used are given in METATOOL[4] format in Appendix A.

➤ **Tricarboxylic Acid Cycle** TCA cycle - this is the same network that have been mentioned in Section 1.5.4. We have chosen TCA cycle (also called *Krebs* cycle [83, 107]) as a simple introductory example of metabolism based on the one designed by Wright et al. [183] in *dictyostelium discoideum*. The version of TCA cycle that we are using for our illustrations was redrawn in [131] (Figure 1.8). This network contains 15 reactions and 11 internal metabolites.

➤ **Mitochondria metabolism**

✓**What is mitochondrion?** Mitochondria are known as the powerhouses of cells. They are very small **organelles** that act like a digestive system that takes in nutrients, breaks them down, and creates energy for the cell. The process of creating cell energy is known as **cellular respiration**. Most of the chemical reactions involved in cellular respiration happen in the mitochondria. A mitochondrion is shaped perfectly to maximise its efforts. We might find cells with several thousand mitochondria. The number depends on what the cell needs to do. If the purpose of the cell is to transmit nerve impulses, there will be fewer mitochondria than in a muscle cell that needs loads of energy. If the cell feels it is not getting enough energy to survive, more mitochondria can be created. Sometimes they can even grow, move and combine with other mitochondria, depending on the cell's needs.

[4] http://pinguin.biologie.uni-jena.de/bioinformatik/networks/

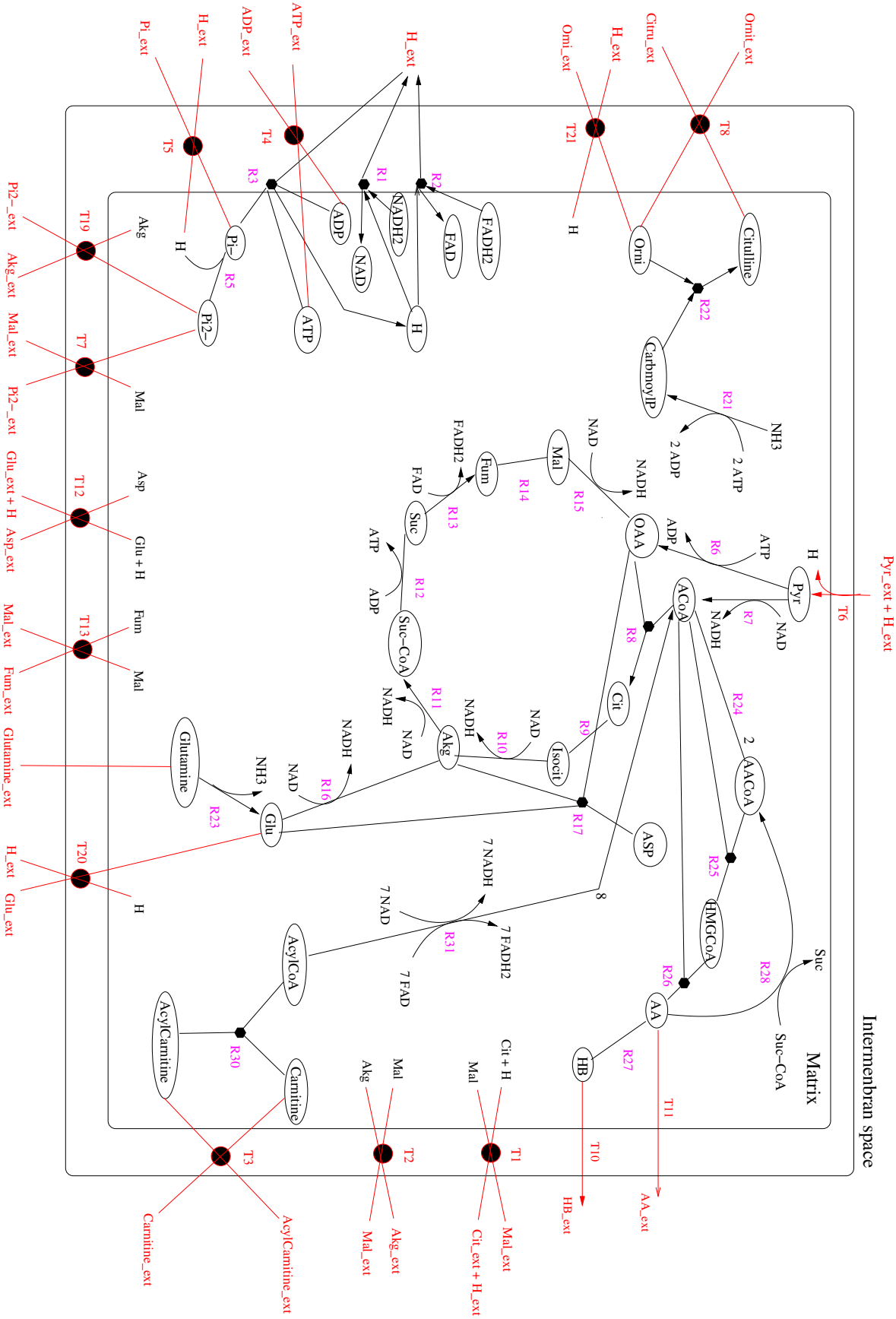


Figure 1.9: General scheme of mitochondrial networks.

✓**Role of Energetic Metabolism** Although some energy can be obtained quickly from glucose or glycogen through anaerobic glycolysis, most of the energy derives from oxidation of carbohydrates and fatty acids in the mitochondria. Energetic metabolism of mitochondria is often described as a set of five main pathways: *TCA cycle*, *respiratory chain*, ketone bodies, beta-oxidation, and a part of ornithine cycle. Depending on the tissues, some variations can be observed. The three retained models concern muscle and liver (*Homo sapiens*) and yeast (*S. cerevisiae*). Both mitochondria of muscle and yeast do not contain an urea cycle. Mitochondria of yeast does not include beta-oxidation as well as production/consumption inside ketone bodies.

To perform the analyses on the metabolic networks of mitochondria, we have chosen 2 of these 3 models: Muscle and Liver. The list of reactions come from the work done in the team for S. Pères thesis [133].

➤ **Metabolic networks of heterotrophic plant cells** Since the metabolic network of heterotrophic plant cell (abbreviated MNHPC) is the main studied object, we have described it more details in Section 4.1 of Chapter 4.

A modified version of MNHPC (called Aracell) adds 12 reactions and 8 metabolites. It is a variation of the plant cell network modified by a biologist who wants to check the consequences of such an addition into the network behaviours.

1.7 Summary

This chapter reviews the context of this research, the basic concepts of metabolic networks. We have also represented the computational models of metabolic networks and the main approaches of studying and analysing them. The last section discussed the data used as experimental materials in our works.

Chapter 2

Network-Based Analysis of Biological Graph

We cannot solve problems by using the same kind of thinking we used when we created them.

Albert Einstein

Complex networks are powerful modelling tool, allowing to study of real world complex systems. They have been used in various domains like computer science, sociology, biology, management, etc. Metabolic network is such a field where a lot of biological processes can be represented in graphical form that is considered as the simplest representation showing the interactions of between metabolites and reactions. The analysis of such networks aims to detect certain properties (e.g. the small-world or the scale-free property) and determine the role of hubs (i.e. highly connected nodes) using some topological measures such as degree distribution, diameter or centralities. Thus, this chapter focuses on the discussion about graphs, global structural properties on graphs, concepts of complex networks and centrality measures.

2.1 Generalities of graphs

2.1.1 Definitions

An *undirected graph* $G = (V, E)$ consists of a set of *vertices* (also called *nodes*) and a set of *edges* (also called *arcs*), where each edge is an unordered pair *u,v* of the vertices. In biological graph, we say there is an edge between *u* and *v* if they are implied in the same reaction (Figure 2.1), without regarding to their directions (i.e. without considering substrate and product, see Chapter 1). Formally, we can define an undirected graph as follows:

Definition 2.1. *A graph is an ordered pair* $G = (V, E)$ *where,*

- V *is the vertex set in which elements are the vertices of the graph. This set is often denoted* $V(G)$ *or just* V*.*
- E *is the edge set in which elements are the edges of the graph, or connections between the vertices of the graph. This set is often denoted* $E(G)$ *or just* E*.*

➤ If the graph is **undirected**, individual edges are **unordered** pairs u, v where u and v are vertices in V. If the graph is directed, edges are ordered pairs (u, v). The **order** of a graph is the number n of vertices in it, often denoted $|V|$ or $|G|$. The **size** of a graph is the number m of edges in it, denoted $|E|$ or G. If $|V| = 0$ or $|E| = 0$, the graph is called *empty* or *null*. If $|V| = 1$ the graph is considered *trivial*.

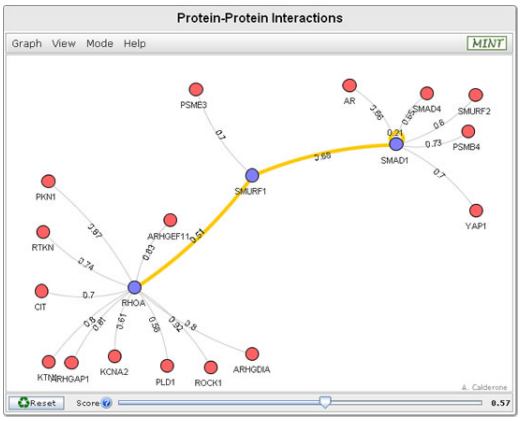


Figure 2.1: Example of interaction graph from Protein-Protein Interactions Browser.
Picture courtesy of Elvevier at
http://www.elsevier.com/about/content-innovation/protein-interaction-viewer

➤ A **directed** graph consists of a set of vertices, denoted *V* and a set of arcs, denoted *E*. Each arc is an **ordered pair** of vertices *(u,v)* representing a directed connection from *u* to *v*. A path from the node *u* to the node *v* is a sequence of arcs (u, u_1), (u_1, u_2), ..., (u_k, v). One can follow such a sequence of arcs to "walk" through the graph from *u* to *v*. Note that a path from *u* to *v* does not imply a path from *v* to *u*. The distance from *u* to *v* is the smallest *k* for which such a path exists. If no path exists, the distance from *u* to *v* is defined to be infinity. If *(u,v)* is an arc, the distance from *u* to *v* is 1.

Graphs are commonly used to represent interactions between proteins. Figure 2.1 shows such a graph where the vertices are proteins and the edges between two proteins are labelled with the interaction name.

Several structural properties can be computed to characterise graph into its applications. We present now the main ones. It is noticeable that the terms of "network" and "graph" will be roughly exchangeable in the context of metabolic networks/graphs.

2.1.2 Global structural properties

The most well-known structural characteristics which are computed point out quantitative evaluation of the size and/or the connectivity density in graph.

➤ **Degree** The most basic measure of a vertex i is probably its *degree* k_i, which is defined as the number of edges adjacent to the vertex [40]. In a network without *self-loops* (i.e. edges that connect a vertex to itself) and *multiple links* (i.e. two vertices are connected by more than one edge), the degree equals to the number of neighbours of the vertex. In the case of **directed** graphs, we distinguish between the *input degree* k_i^{in} and the *output degree* k_i^{out}. In

terms of the adjacency matrix A, the degree of node i is just the some of the ith row of A,

$$k_i = \sum_j a_{ij} \tag{2.1.1}$$

➤ **Degree distribution** One can ask for the probability $P(k)$ that the degree of a randomly chosen vertex equals k. The *degree distribution* [26, 89, 115], $P(k)$ expresses the fraction of the number of vertices in a network G which the degrees equal k. The degree distribution can be calculated by

$$P(k) = \frac{\delta_k}{n} \tag{2.1.2}$$

where δ_k denotes the number of vertices of the degree k in the graph G and n denotes the size of G (the number of vertices of the graph G).

➤ **Distance** The *distance* between any two nodes i and j in the graph G, denoted d_{ij}, is the *length of the shortest path* between the vertices, that is, the minimal number of edges that need to be traversed to travel from i to j. The shortest path between two vertices does not have to be unique, often there exist several alternative paths with identical path length. For directed graphs, the distance between two vertices i to j is usually not symmetric $d_{ij} \neq d_{ji}$. Likewise, for directed, as well as *disconnected graphs*, that is, graphs consisting of two or more isolated components, there might not always be a path that connects vertex i to j. In such a case, the distance between the respective vertices is infinite $d_{ij} = \infty$.

➤ **Average or Characteristic Path Length** The *average path length* or *characteristic path length* or *average distance*, denoted l, is the path length averaged over all pairs of vertices [165]. This parameter measures the typical separation between any two vertices in the graph [179]. This property seems having a similarity to the aspect of diameter which calculates the number of edges in the shortest path among any pair of vertices of the graph G.

For a connected graph G (i.e. existing a path between every pair of vertices), the average distance is given by

$$l = \frac{\sum_{i,j \in V} d_{ij}}{n(n-1)} = \frac{\sum_{i \in V} \sum_{j \in V \setminus \{i\}} d_{ij}}{\binom{n}{2}} \tag{2.1.3}$$

➤ **Diameter** The *diameter* d of a graph G is defined as the maximum distance of any pair of vertices in G, i.e. $d = \max(d_{ij})$.

This fact was stylised in the famous play of John Guare titled "six degrees of separation" that was originally set out by Frigyes Karinthy in 1929. Stanley Milgram pioneered the study of path length through a clever experiment in which people had to send a letter to another person who was not directly known to them of the graph [116].

This property gives the name small-world (Section 2.2.1) to graph applications, because it is possible to connect any two vertices in the graph through just a few links, and the local connectivity would suggest the graph to be of finite dimensionality. Thus, the diameter of a graph tells us how "big" it is, in one sense (that is, how many steps are necessary to get from one side of it to the other).

2.1.3 Computing global structural properties of concrete networks

In order to see the influence of the structural properties on how to characterise networks, we have computed them on the different metabolic networks described in Section 1.6.

➤ **Building reaction and metabolite networks** From the first set of the 5 networks: TCA cycle, Mitochondria Muscle and Liver, and the two metabolic networks of heterotrophic plant cell, we have built several graph-based representations: complete network, reaction network and metabolite network. Complete networks have been built as directed graph, reaction and metabolites as undirected networks because most of reactions are reversible.
In the complete network where both reactions and metabolites are vertices, it exists an edge between one metabolite and a reaction if this metabolite is implied in the reaction. The method used for building the reaction and metabolite networks was proposed by Wagner and Fell [177] and reused in [106]. The detail of the method is to extract reaction and metabolite networks from the complete one is as follows:

✓ The reaction network is an ordered pair $G_R = (V_R, E_R)$ where the vertex set V_R consists of all chemical reactions in the network and E_R the edge set. Two reactions R_1, R_2 are adjacent if it exists an edge $e = (R_1, R_2) \in E_R$, i.e. they share at least one chemical compound (metabolite), either as substrate or as product.

✓ The metabolite network is an ordered pair $G_M = (V_M, E_M)$ where the vertex set V_M consists of all chemical compounds (*metabolites*) belonging to the network and E_M the edge set. Two metabolites M_1, M_2 are adjacent if there exists an edge $e = (M_1, M_2) \in E_M$, i.e. they occur (either as *substrates* or *products*) in the same chemical reaction.

➤ **Results:** Before evaluating the results obtained for all networks, we want to remind that TCA cycle is a single pathway, the two mitochondria networks contain several pathways and the plant cell networks are scaled at the cell metabolism level even they do not include a full one (see Section 1.6 of Chapter 1). Table 2.1 shows the obtained results for all networks from the smaller to the bigger one. This result was presented at 71st Harden Conference Metabolic Pathway Analysis (UK, 2011) [112].

✓ The column 2 and 3 in the table depict the **number of vertices and edges** respectively. Obviously, the complete networks have more vertices than the other ones. Meanwhile, it is

worth to note that the reaction networks, the second group in the table, have more edges than the corresponding complete one (excluding *TCA cycle react*, probably because it is too simple to illustrate the case). Especially, the *Aracell react* has 3 times more edges than the *Aracell complete*, that suggests that *Aracell react* is more packed than the others. In the last group, the metabolite networks, the first three ones own too a greater number of edges than the complete networks while the *MNHPC meta* and *Aracell meta* do not.

Table 2.1: Computing the global structural properties of some example networks. The number of vertices, the number of edges, the average degree, the average path length and the diameter are computed in 3 groups of networks (e.g. complete, reaction and metabolite network).

Species	Nb. V.	Nb. E.	Avg. Deg.	Avg. P. L.	D.
Complete networks					
TCA cycle	40	63	3.150	3.537	8
Muscle	89	161	3.618	4.135	10
Liver	105	191	3.638	4.303	12
MNHPC	148	217	2.932	5.247	12
Aracell	170	282	3.337	4.843	12
Reaction networks					
TCA cycle react	15	44	5.867	1.657	3
Muscle react	37	225	12.162	1.809	4
Liver react	44	288	13.091	1.911	5
MNHPC react	78	441	11.308	2.414	5
Aracell react	91	793	17.429	2.231	5
Metabolite networks					
TCA cycle meta	25	94	7.520	1.900	4
Muscle meta	52	223	8.577	2.282	5
Liver meta	61	262	8.590	2.368	6
MNHPC meta	70	181	5.171	2.865	6
Aracell meta	78	264	6.769	2.644	6

✓ The column 4 in the table depicts the **average degree** of the vertices. The average degree in the case of the complete networks is pretty similar although the network scale is biologically different, from pathway to cell (as we have mentioned previously). In addition, we have the same observation on both the reaction and metabolite networks. That is probably because the network structure influences on the average degree rather than the network size. Even though the average degree increases roughly between 3 groups depending on the number of edges, it is clear that no direct correlation exists between the average degree and the number of edges. To further investigate what happens, we have visualised the histogram of degree distribution of the 3 networks groups (Figure 2.2).

✓ In Figure 2.2, the **degree distribution** of the 5 complete networks reveals the well-known fact that most of the metabolic reactions are in the form of "2 substrates give 2 products" or "1 substrate gives 1 products" as shown in Figure 2.2a. The histogram exhibits two peaks around 2 and 4 degrees. Unlike those degrees, the higher ones (≥ 9) concern about 10% of the total number of vertices for all the complete networks. In the same way, both histograms of reaction and metabolite degree distributions show that more than 10% of reactions or metabolites can be considered as ***"hubs"***. This concept is wide-spread for metabolites but not really for reactions. It suggests that a group of reactions can control a lot of processes. Finally, interpretation of average degree is found to be tricky and not appropriated as it is, to study the structure of our networks.

✓ The **characteristic path length** and **diameter** are depicted in the columns 5 and 6. The complete networks exhibit the highest values. This is directly linked to their higher number of vertices. Meanwhile, the reaction and metabolite networks have pretty the same range of values. Again no simple explanation can be given for that.

As pathways which compose metabolic networks are often considered as modules in such networks. We have tried to work with the clustering coefficient that is a technique to check modularity through networks.

2.1.4 Checking network modularity

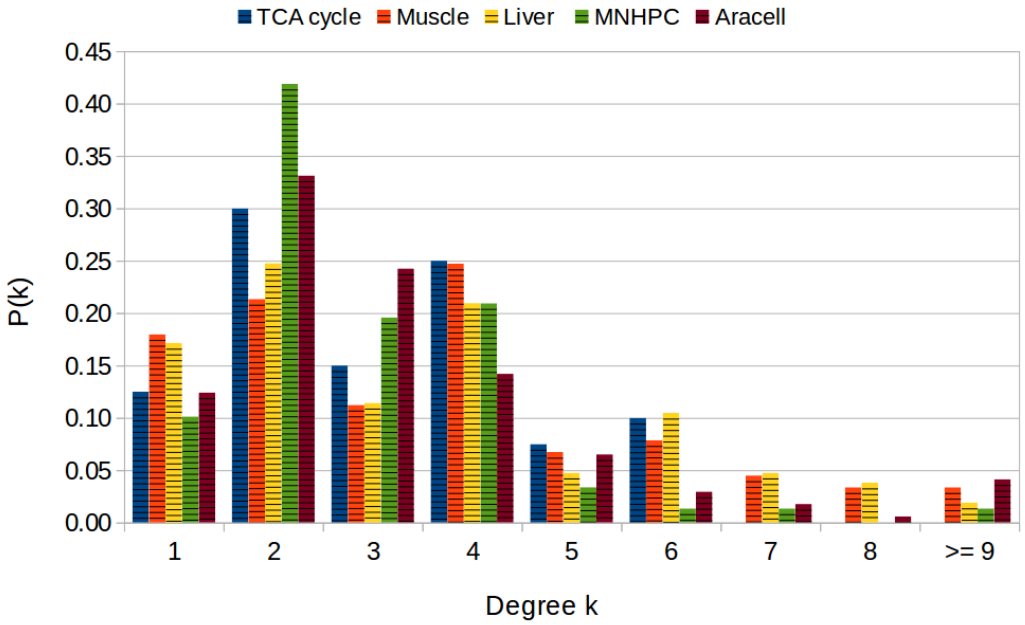
Extracting characteristics of network structure could be achieved by computing modularity parameters as the clustering coefficient.

➤ The *clustering coefficient* C, based on Watts' proposal [179], is a measure of the cliquishness of the local neighbourhoods. It represents the probability that two neighbours of a given node are themselves connected. In the case of undirected networks, given a vertex i with k_i neighbours, there exist $E_{max} = k_i(k_i - 1)/2$ possible edges between these neighbours.

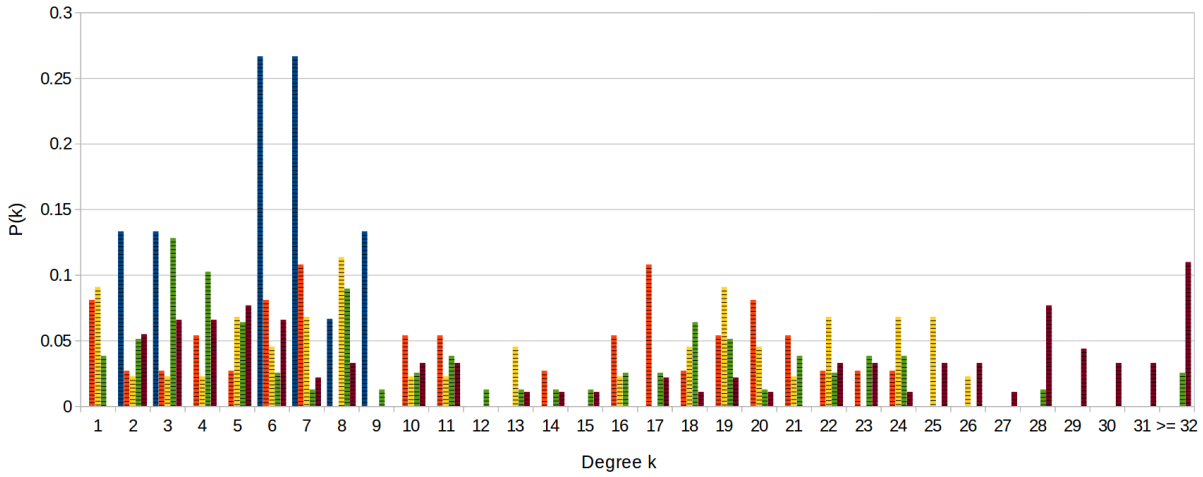
The clustering coefficient C_i of the vertex i is given as the ratio of the actual number of edges E_i between the neighbours to the maximal number E_{max}, by the following equation:

$$C_i = \frac{2E_i}{k_i(k_i - 1)} \tag{2.1.4}$$

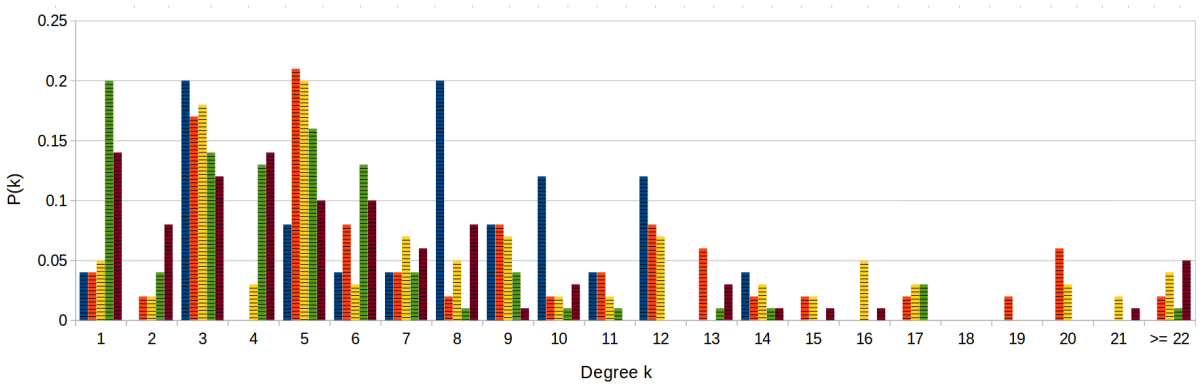
for all $i \in V$. Equation (2.1.4) tells us the frequency of a node's neighbours which are neighbourhood of each other. The calculation of this clustering coefficient is refereed as *local clustering* that differs from *global clustering* [80]. There are a variety of the ways that clustering has been measured. For more discussion on clustering and some empirical examples, see [152]. Here the global or mean clustering coefficient C of the network G, therefore, is defined by the average



(a) Degree distribution of the 5 complete networks.



(b) Degree distribution of the 5 reaction networks.



(c) Degree distribution of the 5 metabolite networks.

Figure 2.2: Degree distribution of the networks

local one of all vertices:

$$C = \sum_i \frac{C_i}{n} \tag{2.1.5}$$

Ravasz et al. analysed the metabolism of *E. coli*. They found that it has a modular topology, potentially comprising several densely interconnected functional modules of varying sizes that are connected by few *intermodule links* [146].

➤ **Results:** we have computed the average clustering coefficients of the 5 complete networks and the results are $zero$ for all. This way to design the networks (with both reactions and metabolites as vertices) is probably not relevant to compute the clustering coefficient.
The clustering coefficients of reaction and metabolite networks are lightly different and seem exploiting functional modules existing. Table 2.2 shows the obtained results for these cases with a value around 0.5, regarding the range of possible values, between 0 and 1 we can consider that some modularities exist in those networks. But not strong information is obtained. For example, once again, we can see in Table 2.2 that TCA cycle network which contains only one pathway has not at all a clustering coefficient really different from the other networks containing several pathways.

Table 2.2: Computing average clustering coefficient distribution of some concrete metabolic networks. The results are computed by using the application *VisANT* [79].

Species	Reaction network	Metabolite network
TCA cycle	0.486	0.601
Muscle	0.639	0.534
Liver	0.645	0.506
MNHPC	0.571	0.336
Aracell	0.611	0.419

In conclusion of this first section about global properties of metabolic networks and about clustering coefficient, roughly speaking, we can see that they cannot be considered as efficient indicators of the network structure in the case of the metabolic networks that we have tested from a very simple one to a cell level one. Thus, as many people have argued that metabolic networks can be seen as complex networks, we have pursued this work by using techniques to analyse complex networks. We will be described them in the next section. One can note that concept of modularity given by the clustering coefficient will be reused in this context.

2.2 Complex networks

As metabolic networks are composition of many biological processes (i.e. metabolic pathways), they are considered as complex networks [186]. Small-world networks (SWNs) and its branch,

scale-free networks (SFNs) are the most well-known classes of complex networks, which are used to model *"real-life"*. This section goes into the aspects of these two classes, they use concepts coming from 2 basic classes of networks: simple networks and random networks. We have assume that most often they are known but if not we have given necessary information about them in Appendices C.2.1 and C.2.2.

2.2.1 Small-world networks

For many real world phenomena, the average path length l of a network is much smaller than that network size n, that is $l \ll n$. Such networks are said to be characterising the small-world property [121, 179]. In mathematics, physics and sociology a *small-world network (SWN)* is a category of networks in which most nodes are not neighbours of one another, but most nodes can be reached from every other by a small number of *hops* or *steps*. D. Watts and S. Strogatz introduced this terminology in 1998 [179] (also called WS model) that was originated from the famous experiment made by Milgram in 1967 [116]. Milgram found that two US citizens chosen randomly were connected by an average of six acquaintances.

➤ Small-world networks in real life

Small-world networks (SWNs) can be found in many real-world applications, including road maps, food chains, electric power grids, metabolite processing networks, networks of brain neurons, voter networks, telephone call graphs, and social influence networks. These systems comprise of many local links and fewer long range *"shortcuts"*, often use with a high degree of local clustering but relatively small diameter (see more detail below). Networks found in many biological and man-made systems are "small-world networks", which are highly clustered, but the minimum distance between any two randomly chosen nodes in the graph is short. Thus, studies on SWNs have been interested by many researchers in a variety of fields such as mathematics, computer sciences, physics, social sciences, etc.

In Goyal's study [63], the principal conditions that a network G exhibits *small-world* properties are as the following:

1. The number of nodes is very large as compared to the average number of links (the average degree), i.e. $n \gg k$
2. The network is integrated; a giant component exists and covers a large share of the population.
3. The average distance between nodes l (called characteristic path length) in the giant component is small, i.e. l is of order $ln(n)$.
4. The global clustering coefficient is high, i.e. $C \gg k/n$

✓ In a study of Indian physicians [159], they have analysed and showed the structure of the Indian railway network (IRN). Identifying the stations as nodes of the network and a train which stops at any two stations as the edges between the nodes, Sen and co-authors measured the

average distance between an arbitrary pair of stations and find that it depends logarithmically on the total number of stations in the country. While from the network point of view this implies the small-world nature of the railway network, in practice a traveller has to change only a few trains to reach an arbitrary destination. This implies that over the years, the railway network has evolved with the sole aim of becoming fast and economical; eventually its structure has become a SWN.

✓ In fact, rich-species food webs with a good taxonomic resolution display the properties of small-world behaviour [117]. Montoya and Solé analysed the four large food webs and compared between real webs and randomly generated webs. Consequently, they approved that the clustering coefficient of both types is the same average number of links per species. One important result is that in all cases, the clustering coefficient is clearly larger than the one of the random networks. For the characteristic path length, the difference between the random and real case is almost very small.

➤ **Properties of small-world networks**

Based on the definition of SWN proposed by [178] and its extensions such as [8, 18, 63], we have described some commonly used properties of small-work networks as follows:

- the network has strong connected components (SCCs).
- the local neighbourhood is preserved (as for regular lattices).
- the diameter of the network increases logarithmically with the number of vertices n (as for random networks).
- the clustering coefficients are much larger than those of the random networks.
- The average length between two points characterising global properties of the network was found to depend strongly on the amount of disorder in the network.

2.2.2 Scale-free networks

According to Barabási et al. [15], a scale-free network is a network whose degree distribution follows a power law. That is, the fraction $P(k)$ of nodes in the network having k connections (also called degree k) follows a well-defined functional form $P(k) \sim k^{-\gamma}$ where the degree exponent γ is a constant whose value is typically in the range $2 < \gamma < 3$, although occasionally it may lie outside these bounds [6].

On the other hand, these scale-free networks own the power-law behaviour means that most vertices are connected sparsely, while a few vertices are connected intensively to many others and play an important role in functionality [61]. Figure 2.3 illustrates the difference between random and scale-free network.

Power-law distribution

A power law is a special kind of mathematical relationship between two quantities. When the number or frequency of an object or event varies as a power of some attribute of that object (e.g., its size), the number or frequency is said to follow a power law. For instance, the number of cities having a certain population size is found to vary as a power of the size of the population, and hence follows a power law [34].

➤ Real phenomena modelled as scale-free networks

Scale-free networks are noteworthy because many empirically observed networks appear to be scale-free, including World Wide Web, Internet, citation networks, biological and some social networks. These networks also behave in certain predictable ways; for example, they are remarkably resistant to accident failures but extremely vulnerable to coordinated attacks. Scale-free networks have been also applied in the power grids, the stock markets and cancerous cells, as well as the dispersal of sexually transmitted diseases (see examples in Table 2.3).

Table 2.3: The real world phenomena modelled as scale-free networks [16]

Network	Node	Links
Cellular metabolism	Molecules involved in burning food for energy	Participation to the same biochemical reaction
Protein-Protein Interactions (PPI)	Proteins to regulate a cell's activities	Interaction among proteins
Hollywood	Actors	Appearance in the same film
Internet [48]	Routers	Optical or other physical connections
Research collaborations	Scientists	Co-authorship of papers
Sexual relationships	People	Sexual contact
World Wide Web [15]	Web pages	URLs

➤ Properties of scale-free network

A variety of complex systems characterised by a power law distribution have similar important properties. Barabási and Bonabeau [16] listed some scale-free characteristics as follows:

- Some nodes, called **hubs**, have highest degree and are thought to serve specific purposes in their networks. The hubs can have hundreds, thousands or even millions of links.
- As scale-free networks are known as robust against accidental failures but vulnerable to coordinated attacks [16].

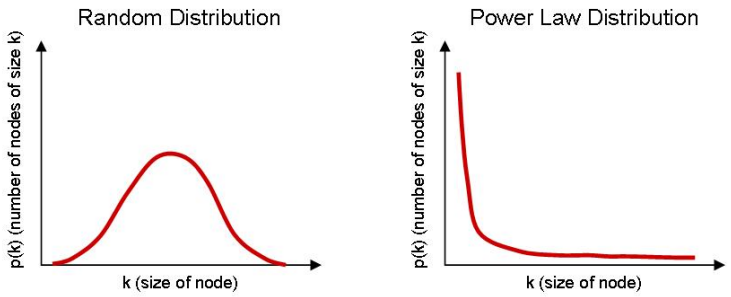
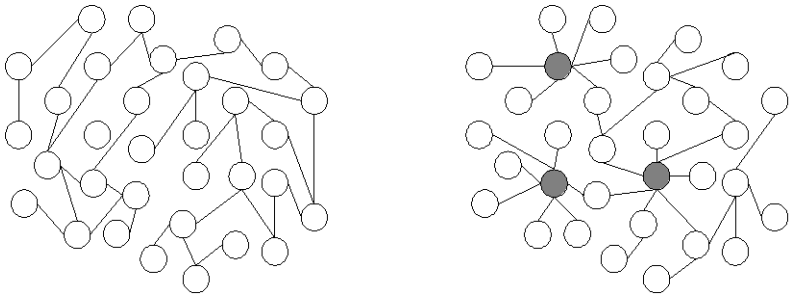


Figure 2.3: Random and power law distribution

– Scale-free networks may show almost no degradation as random nodes fail. Connectivity in the network is maintained thank to the hub nodes. Thus, if there is some connected troubles, the network may still work.

– In a targeted attack, in which failures are not random but are the directed results at hubs, the scale-free networks would be failed catastrophically.



(a) Random network (b) Scale-free network

Figure 2.4: Random network (a) and scale-free network (b). In the scale-free network, the larger hubs are highlighted.

2.2.3 Metabolism as a complex network

Complex networks discipline studies relationships between parts to the collective behaviours of a system and how the system interacts and forms connections with its environment. As mentioned in previous sections, metabolic networks are made up by complex biological processes and they can be considered as complex dynamic systems. Hence, we present several main concepts used to analyse complex networks.

➤ Metabolism reveals small-world properties

There exist many phenomena using small-world networks to represent their interrelated components. Watts and Strogatz showed that several biological, technological and social networks are of the small-world type [179], whereas Wagner and Fell [50, 177] proposed theory and methods to analyse the structure of the *E. coli* metabolism modelled as small-world network. They evolved that there is no very faithful representation of metabolism would be performed by completely random networks. In addition, protein complexes can be represented as SWN, exhibiting a relatively small number of highly central amino-acid residues occurring frequently at protein-protein interfaces [39]. The representation of protein structures as SWN has recently become an interesting approach to study a variety of problems associated to protein function and structure, such as the identification of key residues involved in the protein folding mechanism [174] and the identification of functional sites in protein structures [9] among other examples.

➤ Metabolism as a scale-free network

In the most fully connected biochemical networks, modular organisation is not apparent [146]. The clear boundaries between sub networks do not show out facilitating to study the relationships among them. The studies in literature also have suggested that metabolic networks in all organisms have potential capabilities to be highly modularised. Another aspect studied in [49] showed that it is possible to use the subgraph extraction to find pathways out from metabolism or biological components such as genes, proteins, compounds, etc. In the paper [35], the authors stated small-world behaviour and efficiency of a network. They also showed that neither random graphs nor small-world networks constructed according to the Watts and Strogatz model, have a power-law degree distribution $P(k)$ like the one observed in real large networks.

Looking the above achievements, we have tried to uncover small-world and scale-free features in our metabolic networks. To be relevant, our example of plant cell network, MNHPC, has been chosen to verify small-world properties because it is complex enough.

➤ Verifying small-world and scale-free properties in plant cell metabolism

The values[1], given in Table 2.4, reveal that MNHPC has the average degree k very less than the number of vertices $n = 148$. This network is strongly connected because it has only one strong connected component. Besides, the characteristic path length of the network l is of order to $ln(n)$. However, the clustering coefficient $C = 0$ is less than the fraction k/n. Consequently, we can state that, in a first attempt, MNHPC does not really satisfy small-world model.

[1] These values are computed using the *igraph* package for R programming language [138]

Table 2.4: Computing the small-world properties of MNHPC complete network. These properties are expected to satisfy the principal criteria suggested by Goyal [63].

Property	Value	Evaluation
The average degree k	2.932	$\ll n = 148$
The network connectivity (the number of SCCs)	1	The network is integrated
The average distance l	5.247	be of order $ln(n) = 4.997$
The clustering coefficient C	0	$< k/n = 2.932/148 \sim 0.0198$

✓ As it has been shown in Section 2.1.3, we have computed the degree distribution of the our 5 networks examples (explained in Section 1.6). The histograms of degree distribution confirm that any feature of power-law distribution (i.e. a core characteristic of scale-free network) can be found. But several another parameters can help to characterise complex networks. The next section presents them and the obtained results with these measures.

2.2.4 Complex networks analysis

Even our metabolic networks seem to not follow exactly the rules given for SWN and SFNs, we have explored the concepts of centrality to verify whether we can obtain some information about the network structure or not.

➤ **Network centralities** Closely related to distance measures, network centrality measures aim to characterise each vertex or edge with respect to their position within the network. We will briefly outline here some basic features of these metrics[2].

Indeed, several studies on biological networks have revealed a significant relationship between vertex degree (as presented previously) and functional importance of vertices [7]. However, the degree is clearly not the only determinant of the functional importance of a vertex. The general question of complex networks analysis problem is to determine the most important (also called central) elements that have better access to information and better opportunities to spread information. The two of the oldest concepts in network analysis are centrality and centralisation, which has used to rank importance level of vertices.

✓ Ranking of objects is usually based on numerical values. A function that assigns a numerical value to each vertex of a network is called a *centrality*. This concept is also called with different names such as centrality measure or centrality index.

Definition 2.2 (centrality)**.** *Let $G = (V, E)$ be a directed or undirected graph. A function $\mathcal{C} : V \mapsto \mathbb{R}$ is called a centrality.*

Centralities allow a pairwise comparison of the vertices, for example, a vertex v_1 is said to be more central or more important than a vertex v_2 if $\mathcal{C}(v_1) > \mathcal{C}(v_2)$.

[2]The examples in [89, Ch.4,p.65] clearly explains different centralities.

In this section, the four concepts of centrality, which based on degree and shortest path, are addressed. The first centrality, degree, is almost trivial as we have seen. It counts the number of edges attached to a vertex. The other three centralities use information about shortest paths between vertices of the network. All these degree-based and shortest path-based centralities are defined for undirected and non weighted networks.

In fact, degree is a local centrality measure. Only the immediate neighbourhood of the vertex of interest is considered. For our networks, it has been shown that metabolites as well as reactions with a high degree value are more likely to be essential for the organism than ones with a lower degree value.

✓**Eccentricity Centrality** First of all, we consider the following example. A map of a city is given, roads are modelled as edges, and vertices represent potential places for a hospital to be constructed within this city. The position for the hospital should be chosen such that it is reachable from all other places with the least moves possible (measured by the shortest path distance).

Definition 2.3 (eccentricity centrality [89])**.** *Let $G = (V, E)$ be an undirected and connected graph. The eccentricity centrality is defined as:*

$$\mathcal{C}_{ecc}(s) := \frac{1}{max\{d_{st} : t \in V\}} \tag{2.2.1}$$

where d_{st} denotes the distance between the vertices *s* and *t*, that is, the length of a shortest path between *s* and *t*.

✓**Closeness Centrality** The closeness centrality can be explained in the same context as the eccentricity centrality. Instead of a hospital a shopping mall has to be placed onto the map. For a shopping mall the constraint is that most customers can reach it comfortably. Therefore it is placed at a point where the shortest path distances for all vertices to the position of the mall is minimised.

Definition 2.4 (closeness centrality [149])**.** *Let $G = (V, E)$ be an undirected and connected graph. The closeness centrality is defined as:*

$$\mathcal{C}_{clo}(s) := \frac{1}{\sum_{t \in V} d_{st}} \tag{2.2.2}$$

✓**Betweenness Centrality** Every vertex that is part of a shortest path between two other vertices can monitor communication between them. Counting how many communications a vertex may monitor leads to an intuitive definition of a centrality: A vertex is central if it can monitor many communications between other vertices.

Let σ_{st} denote the number of shortest paths between two vertices *s* and *t* and let $\sigma_{st}(v)$ denote the number of shortest paths between *s* and *t* that use *v* as an interior vertex. The rate of communication between *s* and *t* that can be monitored by an interior vertex *v* is denoted by $\delta_{st}(v) := \sigma_{st}(v)/\sigma_{st}$. If no shortest path between *s* and *t* exist ($\delta_{st} = 0$), then we set $\delta_{st}(v) := 0$.

Definition 2.5 (shortest path betweenness centrality [10, 57, 58])**.** *Let $G = (V, E)$ be an undirected network. The shortest path betweenness centrality is defined as:*

$$\mathcal{C}_{spb}(v) := \sum_{s \in V \wedge s \neq v} \sum_{t \in V \wedge t \neq v} \delta_{st}(v) \tag{2.2.3}$$

2.2.5 Experiments: Finding high-centrality hubs

While the degree k of a node explains the general topological features of the network and can only capture the local structure of network nodes (nearest neighbours), the betweenness centrality C_{spb} of a given node i is related to how frequently a node occurs on the shortest paths between all the pairs of nodes in the network (see previous definition). Hence, betweenness centrality identifies nodes with great influence over how the information reaches distant network nodes. This metric has been used to measure the global relationships of drug-therapy interactions [118], and to detect essential proteins and their evolutionary age [84], to model epidemics, for identifying key players in spreading an infection [126] ...

➤ We have run these algorithms on our own data. In Tables 2.5 and 2.6 we expose the top-20 nodes (i.e. reactions and metabolites) with highest betweenness in the MNHPC reaction and metabolite networks. This information is associated with the result of the *closeness* centrality C_{clo}, which measures how close a given node i is to others [149]. For each measure, the maximum is given in red colour.

- One can see in the reaction network that, even the two first values of betweenness, reactions `Vhk1` and `Vhk2`, are closed for degree and closeness centrality, interpretation of eccentricity and degree ranks is not directly given. In addition, these two reactions are energy reactions, their centrality is not really surprising. In the case of the rest of the list, the obtained values are not correlated.

- In the metabolite network, obviously metabolites like `ATP`, `DHAP_p`, `NADH`, `CO2`, `NADPH` are placed in the top ranking of the result because they are commonly taken part in the metabolic processes. This fact is prominent. Again, the order of the rest of the metabolites appears not directly linked.

➤ As we have discussed, the closeness centrality can be understood as a measure of how long it will take for information to spread from a given node to distant nodes in the network. Thus, nodes with high closeness indicate that their influence can reach others more rapidly. We

can see in Table 2.5, no valuable information can be extracted from the closeness values in the case of the reaction network. In the other hand, Table 2.6 shows the metabolites which are at the top-20, a well-known information. As we want to characterise, for example, dependencies between nodes through the different pathways, it seems that such information is hard to extract.

Table 2.5: Top-20 reactions with the highest betweenness in MNHPC reaction network. The computing results of degree centrality, closeness centrality, betweenness centrality, eccentricity and node degree are too displayed.

Vertex	Closeness Centrality	Betweenness Centrality	Eccentricity	Degree
Vhk1	0.535	0.109	4	23
Vhk2	0.542	0.096	4	24
Vgapdh_p	0.592	0.084	3	33
Vg6pdh	0.517	0.078	3	18
Vglyc3P	0.588	0.074	5	32
Vcl	0.542	0.067	3	24
Vrbco	0.513	0.059	3	24
Vpk	0.542	0.057	4	23
Vala	0.430	0.054	3	11
Vgs	0.478	0.051	3	17
Vpfk	0.527	0.043	3	21
Vg6pdh_p	0.484	0.041	4	17
Vme	0.494	0.038	4	18
Vat	0.510	0.037	4	19
NRJ1	0.570	0.034	5	28
Vasp	0.433	0.032	4	11
Vinv	0.381	0.03	3	8
Vsps	0.381	0.026	4	8
Vpgk	0.513	0.026	4	19
Vidh	0.503	0.024	4	18

2.2.6 Community detection and Subgraph extraction

Social networks are examples of graphs with communities. The word community itself refers to a social context. People naturally tend to form groups, within their work environment, family and friends [54]. Relationships/interactions between elements of a biological graph can be formed groups which tend to share common behaviours or characteristics. Extraction of such communities is a big challenge in the graph theory. Scientists working in Bioinformatics field are interesting to solve this problem in the context of metabolic networks. Using seed nodes in the network to predict pathways, Helden and co-workers [49] have tried to extract subgraphs to any biological networks. They comparatively evaluated seven sub-network extraction approaches on 71 known metabolic pathways from *Saccharomyces cerevisiae*. The best performing approach is a novel hybrid strategy, which combines a random walked-based reduction of the graph with a shortest paths-based algorithm, and which recovers the reference pathways with an accuracy of $\sim 77\%$ [77]. This method is mainly based on the bow-tie connectivity structure

Table 2.6: Top-20 metabolites with the highest betweenness in MNHPC metabolite network. The computing results of degree centrality, closeness centrality, betweenness centrality, eccentricity and node degree are too displayed.

Vertex	Closeness Centrality	Betweenness Centrality	Eccentricity	Degree
ATP	0.580	0.494	3	29
DHAP_p	0.451	0.119	4	14
NADH	0.489	0.101	4	17
CO2	0.469	0.097	4	17
NADPH	0.466	0.092	4	13
Glc	0.413	0.086	4	6
pyr	0.454	0.067	4	9
Fru	0.404	0.064	4	6
G6P	0.445	0.063	4	8
OAA	0.457	0.059	4	10
ala	0.337	0.058	5	5
glu	0.394	0.051	5	9
Ru5P	0.379	0.049	4	5
F6P	0.413	0.048	4	6
aKG	0.404	0.044	5	11
mal	0.375	0.037	5	6
UDPG	0.315	0.034	5	7
cit	0.399	0.033	4	5
Suc	0.311	0.032	5	5
ADPG	0.377	0.029	4	3

using a *distance definition* derived from the path length between two reactions. This theory combines the properties of the global network structure and local reaction connectivity rather than, primarily, based on the connection degree of metabolites. He asserted that metabolic networks have typical characteristics of small-world networks, namely a power law connection degree distribution.

Unfortunately, it looks our examples not to exhibit the same properties and moreover, we have not found any way to describe efficiently the constraints that we want to take into account by using these network models. We have no weight on the edges and it is difficult to translate the metabolic constraints like: *"all the internal metabolites have to be balanced"* in these kind of models. As it exists specific methods dedicated to metabolic networks after verifying that the most known ones in graph theory do not really provide best results than the specific ones, we have chosen to focus on these methods: EFMs which have been mentioned in chapter 1 and computing minimal cut sets which will be described in the next chapter.

2.3 Conclusion

In this chapter, we have provided a review of graph and network global structural properties which are used in graph theory. The computation of these properties have been performed on the several complete networks as well as the reaction and metabolite networks. Two another models

of complex networks have been described - small-world network and scale-free network models. The centrality measures were studied to determine the most central nodes. It is said that: degree distribution, diameter, average path length, eccentricity, closeness centrality and betweenness centrality, which are the inherent properties of complex networks, play the important role of identification organisational hubs trough the networks. Even authors as Fell et al. [50] show that metabolic networks could be considered as small-world networks, the examples that we have concretely studied do not exhibit values useful to extract new information about their structures.

Summing up, the analysis of real-life complex systems as well as metabolic networks poses a number of new challenges, that make us having to combine the different theoretical approaches. Moreover, the remarkable lack of a few generalised small-world behaviours can, as in the MNHPC case, be explained that we have just a partial view of the complete system [109]. That is one of the reasons why we want to move to the other method (*computing Minimal Cut Sets*) that will be presented in Chapter 3.

Chapter 3

Computing Minimal Cut Sets

> *The best programs are written so that computing machines can perform them quickly and so that human beings can understand them clearly. A programmer is ideally an essayist who works with traditional aesthetic and literary forms as well as mathematical concepts, to communicate the way that an algorithm works and to convince a reader that the results will be correct.*
>
> **Donald Ervin Knuth**
> — Selected Papers on Computer Science

In Chapter 2, by calculating the global structural properties and centrality measures, we have known coherent and relational characteristics of metabolic networks. In other words, there are metabolites as well as reactions playing the tremendous roles and others are used as additive elements. In this chapter, the affects of these elements can be recognised more and more in case of discovering set of links or nodes which removals disconnect the network. This technique is called as computing minimal cut sets (MCSs) that used to be a key part in the research. In the beginning of this chapter, the basic concepts of minimum cuts in graph theory will be given. After that, the concepts of MCSs and their applications in biology context will be presented in the following sections.

3.1 Minimum cuts in graph

Whitney [181] is one of the precursors who used the concept of cut sets with planar graphs in the early 1930s. This field, however, had been fallen in oblivion for a long time until the emergence of the modern complex networks theory in the 1960s. Several researches into the theory of the reliability and survivability engineering in complex networks can be found in [23, 141, 187]. In this theory, the problem can be stated informally that if there are several edges of a network failed with an certain probability, disconnection of the network is at a minimum cut. Likewise, minimal cuts have also been arisen in communication networks [139], in information retrieval [27], in compilers for parallel languages [31], and in routing of ATM networks [184]. In systems biology,

minimum cut sets method has been employed into studying the way to stop the production of some interest metabolites in metabolic networks [100]. Therefore, we shall now present the main principles of minimum cuts in graph theory in the next sections.

➤ **Minimum cuts in undirected graph** Let $G = (V, E)$ be an undirected graph. Formally, we define that a *cut* C of an undirected graph G is a partition of the vertices $V(G)$ into two separate non-empty subsets, that is, $C = \{S, \overline{S}\}$ where $S \cup \overline{S} = V(G)$ and $S \cap \overline{S} = \emptyset$ [40]. The set $\delta(S) = \{(u, v) \in E : u \in S, v \in \overline{S}\}$ is a *cut set* since their removal from G disconnects G into more than one subgraphs.

The size of the *cut* C is defined as the number of the edges (in the case of a unweighted graph) or the sum of the weights of the edges (in the case of a weighted graph) in $\delta(S)$. Thus, it can be said that a *minimum cut* is a cut of the certain minimal size. Accordingly, the edges set crossing that minimum cut is called a *minimum cut set*. For an illustrative example, consider the undirected and weighted graph in Figure 3.1. In this example, $(\{a, b, d, e\}, \{e, g, f\})$ is a minimum cut (the bold line) and the minimum cut set corresponding with the minimum cut set is $\{(b, c); (e, f)\}$ whose weight is 9.

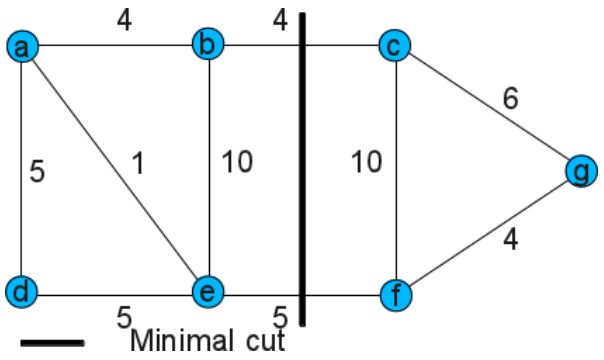


Figure 3.1: A minimum cut of an undirected graph G

➤ **Minimum cuts in directed graph** Similarly, we define minimum cuts in a directed graph [69]. We denote $G = (V, E)$ a directed graph (or a *digraph* for short) with a vertex set V and an edge set E. A minimum cut, like in an undirected graph, is a partition of the node set V into two disjoint subsets. A minimum cut set of G corresponding to that minimum cut is a set of all the edges crossed through these two subsets. However, one should pay attention to how to compute minimum cut value. Instead of summing the weights of all the edges, the only crossed edges between the two subsets coming out S are taken into account. For the directed and unweighted graphs, the minimum cut value is defined as the number of the edges inside that cut set.

3.1.1 Concepts of s-t cut

Practically, we have been working a lot of graph-based complex systems having more than one inputs and outputs. A natural question can be risen whether or not we can cut such graphs into two separate parts containing the inputs S (*sources*) and outputs T (*targets*) respectively. In this context, we often use the concept $s-t$ *cut* with two special terminals: one source node called s and one target node called t [166]. General speaking, the $s-t$ cut [13] is a cut with s and t in different partitions. Formally, a cut $s-t$ of an undirected graph G is simply a cut $C=\{S,\overline{S}\}$ with $s\in S$ and $t\in\overline{S}$. So, a cut set of the $s-t$ cut, denoted by $\delta(S_{s,t})$, is the edge set which end points are in the separate subsets of the vertices. The removal (or "cut") of the edges out of $\delta(S_{s,t})$ disconnects the graph into two separate subgraphs.

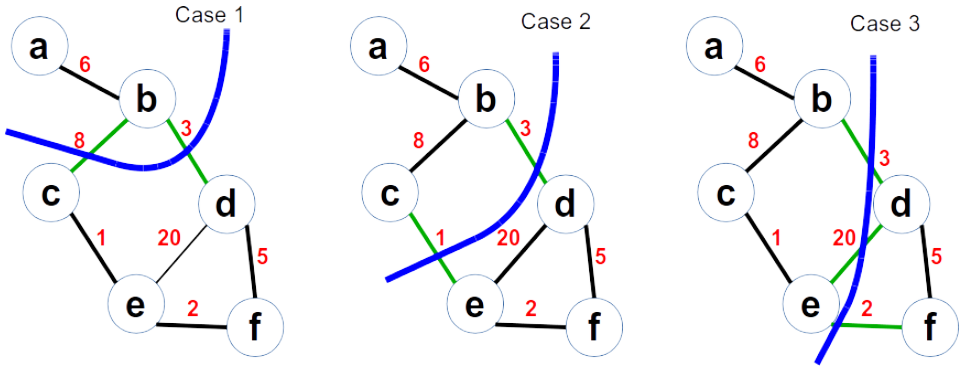


Figure 3.2: Examples of a s-t cuts in undirected graphs.

Let us consider the example in Figure 3.2. Suppose that the $s-t$ cut set of $s=a$ and $t=d$. We can enumerate several $a-d$ cut sets such as $\{bc,bd\}$ with the weight of 11 (in Figure 3.2 it is case 1), $\{ce,bd\}$ with the weight of 4 (i.e. case 2), or $\{bd,de,ef\}$ with the weight of 25 (i.e. case 3). Because of the less number of cut sets, the enumeration can be done in manual. The minimum cut set is the one with the minimum weight (i.e. it is 4 - case 2).

3.1.2 Minimum cuts algorithms

It exists a lot of algorithms for finding as well as enumerating all minimum cut sets of an arbitrary graph. Theoretical algorithms for computing minimum cuts in graphs were proposed from 1961 which can be listed here like Gomory and Hu [62], Hao and Orlin [68], Nagamochi and Ibaraki [119], Stoer and Wagner [164]. In the 1970s, the algorithms for computing minimal cut sets, which employed in reliability engineering, were proposed and proved their correctness formally. For example, Ariyoshi proposed a new computing cut sets method by defining a cut set graph with respect to a given graph [11] or Arunkumar and Lee devised an approach concerning with the enumeration of $s-t$ minimal cut sets [12]. Then many authors suggested their new and improved algorithms on various types of graphs such as the efficient enumeration algorithm generating all minimal cut sets separating a special vertex pair in an undirected graph based on

a blocking mechanism [3], the new algorithm based on a subset method and an iterative process to determine all minimal cut sets for all nodes [81]. There were also innovative techniques based on the construction of a *dual graph* from the original one devised by Shen [161] or developed from the algorithm *maximum adjacency search* to find an arbitrary minimum $s-t$ cut proposed by [164] or [137]. Recently, with the development of high performance computing, several methods for solving the cut set problem have been emerging in new domains such as finding the way to stop the production of a certain product in metabolic networks [99] or cutting an image into several segments aims to facilitate in treatment [46]. More details of these algorithms can be found in Appendix C.3.

➤ **Testing minimal cut algorithms** We have applied the above discussed algorithms into several test cases using open source graph library packages such as boost[1], LEMON[2], jgrapht[3], etc. The Stoer-Wagner algorithm is implemented in *boost* graph package (C++) and *jgrapht* (Java) graph framework. Gomory-Hu and Hao-Orlin algorithms are realized in *LEMON* library. The implementations of these algorithms do not provide any proper solution for finding all minimal cuts (or minimum cut sets) of a **directed graph** which the edges have **no weights**.

3.2 Minimal Cut Sets in Metabolic Networks

This section gives us a great insight about the concepts and the applications of Minimal Cut Set (MCS) in metabolic networks. Broadly speaking, MCS is the main concept that serves as a key approach in this research.

3.2.1 Introduction

The theory of MCS has been found in structural studies of biological networks for recent years, which originated from Klamt et. al [100]. This topic has been interested to many researchers in modelling flavonoid metabolism [151] or in studying strategies blocking growth of the central *E. coli* metabolism [176].

In general, Metabolic Pathway Analysis (MPA) identifies the topology of cellular metabolism based on only the stoichiometric structure and thermodynamic constraints of reactions where kinetic parameters are not explicitly revealed and/or required for the calculations [32, 33, 157, 158]. MCS concept has been developed from EFMs computing, an MPA method using convex analysis to identify all possible and feasible metabolic routes for a given network at the steady state (cf. Section 1.5.4). Computing MCSs of a metabolic network consists of finding all reactions sets which removal makes disconnected the biological functions. For example, one can compute: *(i)* MCSs that block growth; *(ii)* MCSs that disable the production of a certain

[1] http://www.boost.org/
[2] http://lemon.cs.elte.hu/trac/lemon
[3] http://www.jgrapht.org/

metabolite; *(iii)* MCSs that block all flux vectors where a undesired compound is produced with a low yield [176].

3.2.2 Defining minimal cut sets of a metabolic network

S. Klamt and E.D. Gilles proposed MCSs concept in the first time in 2004 [100] as follows:

> *"We call a set of reactions a cut set (with respect to a defined objective reaction) if after the removal of these reactions from the network no feasible balanced flux distribution involves the objective reaction."*

...

> *"A cut set $\mathcal{C}$ (related to a defined objective reaction) is a minimal cut set (MCS) if no proper subset of $\mathcal{C}$ is a cut set."*

It exists a distinction between the definition of cut sets in graph theory and these ones. As we have seen before in traditional graph theory, a cut set partitions a graph into two separate parts, whereas Klamt just tells that he wants to cut a route throughout the graph. Instead of cutting a graph into two even more parts, cut sets in metabolic network context mention about stopping the capability of reaching to feasible balanced condition of non-decomposable pathways. In that case, cut sets divides metabolic networks into several separate parts that makes the pathways not touching the objective function.

➤ **The initial concept of MCSs** The algorithm for computing MCSs was proposed by S. Klamt and E. D. Gilles [100], which based on EFMs computing [59, 154, 168]. The idea behind the algorithm for calculating MCSs is the fact that an EFM is the minimal, unique and non-decomposable set of the reactions (enzymes) operated at the steady state; thus removing a reaction from the set in the network prevents to achieve a steady state with the remaining reactions of the EFM. In fact, EFMs and MCSs complement each other, as will be discussed later on.

In biology context, we currently identify the objective reaction for the network function of interest, and EFMs are used for calculating feasible routes for it. Meanwhile, MCSs would be the reactions that cause the dysfunction of these routes with respect to the objective reaction, and so the corresponding network function is stopped.

➤ **Example Network to illustrate MCS algorithm** To illustrate the MCS concept, consider the example network (named *NetEx* used in [99]) and shown in Figure 3.3.

The *NetEx* has some features as follows:

- The network consists of five internal metabolites and eight reactions, of which `R4` and `R5` are reversible;

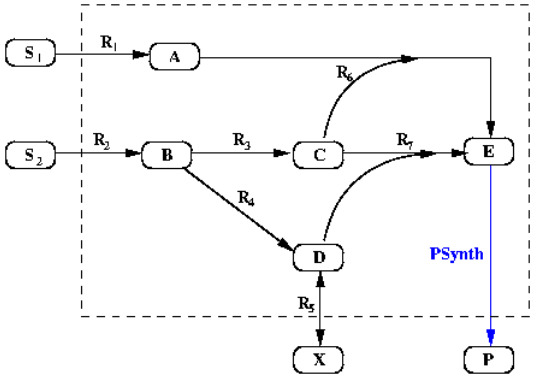


Figure 3.3: Network layout for an example network (*NetEx*) discussed in [99]

- Reactions crossing the system boundaries are coming from/leading to buffered metabolites;
- Assume that the synthesis of product P attracts our attention, hence, all flux vectors with a non-zero flux through reaction `PSynth` are of special relevance for us. Klamt and Gilles [100] called such a reaction of interest *objective reaction*. It is also called *target reaction* in a similar context [182].

3.2.3 Determining MCSs

The MCS algorithm devised by Klamt and Gilles [100] relies on the fact that:

- any feasible steady state flux distribution in a given network, expressed by a vector of the net reaction rates, r, can be represented by a non-negative linear combination of EFMs as illustrated in Equation (3.2.1) (reused from [99]):

$$r = \sum_{i=1}^{N} \alpha_i E_i, (\alpha_i \geq 0) \tag{3.2.1}$$

- where N is the number of EFMs.
- the removal of reactions from the network results in a new set of EFMs constituted by those EFMs that do not involve the deleted reactions.

Before MCSs are computed, the set of $EFMs$ is split into two disjoint sets:

- the set of target modes (EFM^t), i.e., all EFMs ($e^{t,j}$) involving the objective reactions t.
- the set of non-target modes (EFM^{nt}), i.e., EFMs not involving the objective reaction nt.

This MCS algorithm can be divided into two phases as follows:

Preparatory phase

(1) Calculate the EFMs in the given networks.

(2) Define the objective reaction `obR`.

(3) Choose all EFMs where the reaction `obR` is non-zero and store it in the binary array `efms_obR`.

(4) Initialise the arrays `mcss` and `precutsets` as follows: Append $\{j\}$ to `mcss` if the reaction j is essential, otherwise to `precutsets`.

Main phase

(5) FOR i = 2 TO MAX_CUTSETSIZE

(5.1) FOR j = 1 TO q

(5.2.1) Remove all sets from `precutsets` where the reaction j participates;

(5.2.2) Find all sets of reactions in `precutsets` that do not cover any EFM in `efms_obR` where reaction j participates. Combine each of these sets with reaction j and store the new preliminary cut sets in `temp_precutsets`;

(5.2.3) Drop all elements in `temp_precutsets` which is a superset of any of the already determined minimal cut sets stored in `mcss`;

(5.2.4) Find all elements retained `temp_precutsets` which do now cover all EFMs and append them to `mcss`. Append all others to `new_precutsets`;

(5.2) IF `isEmpty(new_precutsets)` BREAK; ELSE `precutsets` = `new_precutsets`;

(6) return `mcss`;

We have rewritten the pseudocode of this algorithm as in Algorithm 3.2.1.

For the *NetEx* network, the algorithm calculates seven MCSs in addition to the trivial MCS (`PSynth` itself). To illustrate, one of the MCSs (MCS6) is shown in Figure 3.4. The eight MCSs and the corresponding EFMs are shown in Table 3.1.

3.2.4 Improvements of MCS concepts

From the original work done by Klamt and Gilles [100], the concepts of MCSs have been generalized and constraint MCSs has been defined some years later.

➤ Generalized concept of MCSs

S. Klamt, in 2006 [99], redefined MCS from that of the original concept expressed under Section 3.2.2, to *"a minimal (irreducible) set of structural interventions (removal of network*

Algorithm 3.2.1: The pseudocode of the MCS algorithm devised by Klamt S. and Gilles S. D. [100]

```
efms ← compute_all_efms();
n ← |efms|; r ← |reactions|; obR ← define_obR(); mcss ← ∅;
efms_obR[i,j] ← 1 otherwise efms_obR[i,j] ← 0 with i = 1..n and j = 1..r;
; // Preparatory phase
for j ← 1 to r do
    i ← 1;
    while i < n and efms_obR[i][j] <> 0 do
        i = i + 1;
    if i > n then
        append(mcss,j);
    else
        append(precutsets,j);
; // Main phase
for i ← 2 to MAX_CUTSETSIZE do
    new_precutsets ← ∅;
    for j ← 1 to r do
        foreach efm in precutsets do
            if j in efm then
                precutsets.remove(efm);
            else
                if not isCovered(efm,efms_obR,j) then
                    efm.add(j); temp_precutsets.add(efm);
        foreach efm in temp_precutsets do
            if isSuperSet(efm,mcss) then
                temp_precutsets.remove(efm);
            else
                if not isCover(efm, efms) then
                    mcss.append(efm);
                else
                    new_precutsets.add(efm)
    if isEmpty(new_precutsets) then
        break;
    else
        precutsets ← new_precutsets;
return mcss;
```

Table 3.1: EFMs and MCSs for the objective reaction *PSynth*. The table shows the EFMs and MCSs of *NetEx*, for the objective reaction *PSynth*

	R1	R2	R3	R4	R5	R6	R7	*Psynth*
Elementary Flux Modes								
EFM1	0	1	0	1	1	0	0	0
EFM2	1	1	1	0	0	1	0	1
EFM3	1	0	1	−1	−1	1	0	1
EFM4	0	1	1	0	−1	0	1	1
EFM5	0	0	1	−1	−2	0	1	1
EFM6	0	2	1	1	0	0	1	1
Minimal Cut Sets								
MCS0								x
MCS1			x					
MCS2	x					x		
MCS3						x	x	
MCS4		x		x				
MCS5		x			x			
MCS6	x			x	x			
MCS7				x	x	x		

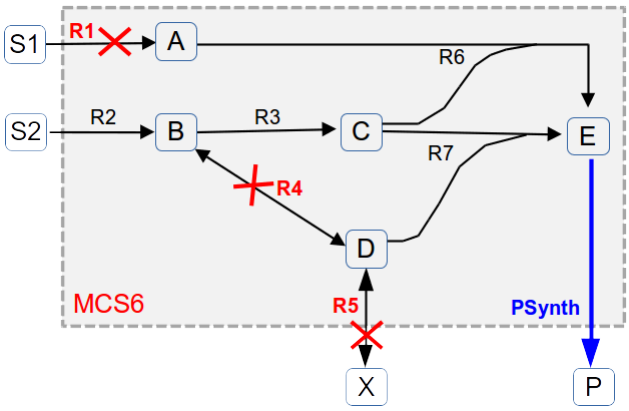


Figure 3.4: One of the MCSs for objective reaction `PSynth`. The simultaneous blocking of reactions `R1`, `R4` and `R5` will eliminate `PSynth` and block the production of `X`.

elements) repressing a certain functionality specified by a deletion task". This new definition is the principal rule that the deletion task plays in the difference between the new generalized approach and the initial MCS concept.

The deletion task can be specified by several Boolean rules that clearly represent and describe, unambiguously, the flux patterns or the functionality to be repressed. This increases the practical applicability of MCSs because they can now be determined for a large variety of complex deletion

problems and for inhibiting very special flux patterns instead of just for studying structural fragility and identifying knock-out strategies [99].

➤ **Constraint MCSs**

To deal with the limitation of stopping desired functionalities along with the targeted reactions, Hädicke and Klamt [66] generalized MCSs to cMCSs that take into consideration of side constraints and allow for a set of desired modes, with a minimum number of modes preserved, to be defined.

As demonstrated in [66], this generalization shows the relationship of the extended approach to Minimal Metabolic Functionality (MMF) (a method based on EFMs computing and was developed by Srienc and coworkers [169, 172]) and OptKnock-related techniques (a group of methods based on the original bilevel optimisation framework with the name OptKnock, developed for suggesting gene knock out strategies for biochemical overproduction [29]). The great flexibility of the new approach is reflected by the fact that popular existing methods such as MMF, OptKnock or RobustKnock[4] can be reformulated as special cases of cMCSs problems.

The refinements and extensions to the initial MCS concept offer a broader range of possible ways in which MCSs can be used to assess, manipulate and design biochemical networks.

3.2.5 Methods to improve MCSs computing

Recent studies [4] have showed hardness of checking that a given set of reactions constitutes a cut. From that we can say finding MCSs for a given set of target reactions is becoming a challenge in large-scale networks. This stems from the fact that almost algorithms for finding MCSs are based on the computing of EFMs with an enormous combinatorial explosion of the number of EFMs [103]. Following the works studied by [73, 86], we point out improvements of MCS algorithms.

We model a metabolic network as a number m of metabolites involved in a set $REACTS$ of r reactions. For our purpose, these reactions can be encoded in a matrix $S(m \times r)$, whose columns encode the metabolites produced and consumed by a given reaction. The matrix S is known as the *stoichiometric* matrix. The reactions may be divided into two types: *reversible reactions*, which can either produce a given output from a given input or vice-versa; and *irreversible reactions*, which cannot operate in reverse.

➤ **Notations** Let REV be the index set of the reversible reactions and $IRREV = REACTS \backslash REV$ be the index set of the irreversible reactions. We call our set of target reactions T; for simplicity, we will usually assume that they are irreversible, that is, $T \subseteq IRREV$.

[4] an implementation of OptKnock

➤ **Mapping EFMs to hypergraph** For the purposes of finding cut sets for a given target T, we consider only the EFMs that include at least one target reaction. Note that cut sets are exactly the sets of reactions that intersect each of these EFMs. The collection of these EFMs constitutes a simple hypergraph (or "Sperner family"[5]) $\mathcal{H} = (REACTS, EFMs)$ on the set of reactions. The key observation is that cut sets are exactly the sets that intersect every edge of $\mathcal{H}$. In the terminology of hypergraphs, such sets are known as *hitting sets* or *vertex covers*. The collection of all minimal hitting sets for $\mathcal{H}$ is itself a hypergraph $\mathcal{H}' = (REACTS, EFMs')$, which is dual to $\mathcal{H}$ in the sense that its minimal hitting sets are the edges of $\mathcal{H}'$. The hypergraph $\mathcal{H}'$ is known as the transversal hypergraph of $\mathcal{H}$ and is denoted Tr($\mathcal{H}$) [73].

➤ **MCSs methods with basing on EFMs computation**

Klamt and Gilles [99, 100] have proposed to first compute the EFMs hypergraph $\mathcal{H}$ via the double description method and then compute Tr($\mathcal{H}$). The computation of Tr($\mathcal{H}$) is done through an enumeration scheme. This method was implemented in the software FluxAnalyzer [100], the predecessor to CellNetAnalyzer [102] that will be presented in the next section. The improved method was suggested by Haus et al. [73] involves modifying existing algorithms to develop more efficient methods for computing MCSs. This improvement was implemented in CellNetAnalyzer [102] preserved for MATLAB environment.

An approximation algorithm for computing the minimum reaction cut and an improvement for enumerating MCSs was recently proposed by Acuña et al. [4]. These emerged from their systematic analysis of the complexity of the MCS concept and EFMs, in which it was proved that finding a MCS, finding an EM containing a specified set of reactions, and counting EFMs are all NP-hard problems.

Jungreuthmayer et al. [85, 86] have developed a new approach to improve the performance of MCSs computing. The idea behind their method is to employ binary patterns.

➤ **MCSs method without basing on EFMs computation**

As we have known, computing EFMs could be a bottleneck in MCSs calculation. Therefore, the methods presented here have tried to avoid computing MCSs via EFMs.

The method based on an algorithm of Fredman and Khachiyan [56] for generating the MCSs directly from the stoichiometric matrix was developed by Haus et al. [73]. The technique is to define a Boolean function that takes a binary pattern of included reactions as input, and yields 1 if this set of reactions is a cut set, and 0 if it is not.

The method, contributed by Ballerstein et al. [14], also determines MCSs directly without computing EFMs. This computational model is based on a dual presentation for metabolic

[5] A hypergraph is Sperner if it has no nested edges.

networks where the enumeration of MCSs in the original network is reduced to identifying the EFMs in a dual network so both EFMs and MCSs can be computed with the same algorithm. They also proposed a generalisation of MCSs by allowing the combination of inhomogeneous constraints on reaction rates.

3.2.6 Computing tools

The next sections discuss at the available tools for EFMs and MCSs computing that we have been used in our works.

➤ CellNetAnalyzer

CellNetAnalyzer (CNA)[6] comes from the previous software Metatool[7] written by the Jena Bioinformatics group. This version was developed in MATLAB containing several modules to visualise and analyse network structures. CNA enables users to compute both EFMs and MCSs. Thus we have used it for calculating EFMs and MCSs of 4 networks. That computation has often been time consuming, in some cases several hours or days are necessary. For example, to obtain MCSs of MNHPC with CNA more than 10 days have been needed with a Linux server, and in the case of Aracell network memory requirements are larger than the amount of memory that the method can manage.

➤ Efmtool and regEfmtool

A couple of years ago, a new implementation of EFMs computation was done by Terzer [167] with improvements of the original algorithm. This is *Efmtool*[8] [167] implemented in Java programming language. *Efmtool* supports multi-threading and seems to be robust to compute large-scale networks. But even this software is freely published under the open source software license *Simplified BSD Style License*[9], this program is not easy for use and is lacking of a detailed documentation. Within recent years, a new software, named regEfmtool[10], derived from the software *Efmtool* and written by C. Jungreuthmayer [87], provides an more easily used tool for computing EFMs with a more complete documentation. They have also proposed a way to define some logical rules to compute EFMs that containing or not some reactions, thereby significantly reducing the size of the obtained solutions and computational costs as well.

The larger networks that we have computed with this tool contain more than 80 reactions. We have obtained several millions of EFMs in only a couple of hours.

[6] http://www.mpi-magdeburg.mpg.de/projects/cna/cna.html
[7] http://pinguin.biologie.uni-jena.de/bioinformatik/networks/
[8] http://www.csb.ethz.ch/tools/efmtool/
[9] http://opensource.org/licenses/BSD-2-Clause
[10] http://www.biotec.boku.ac.at/regulatoryelementaryfluxmode.html

➤ **mcsCalculator**

Computing MCSs out of *MATLAB* and in C language will be soon available from the same team. In preliminary tests, we have been able to obtain MCSs that have not ever been obtained before with *MATLAB* programs due to overload memory. Via communicating personally, we have tested, verified and used *mcsCalculator* for computing MCSs in our data networks.

3.3 Experiments

In this section, we present the results obtained by computing MCSs in several real datasets. The chosen datasets have been described in Section 1.6: mitochondria tissues and heterotrophic plant cells. The purpose of the computation is to verify the hypothesis whether MCSs provides an easier approach to analyse metabolic pathways or not.

3.3.1 Contrast in EFMs and MCSs results

To follow the argument of the authors of MCSs methods, we have tested with different network sizes the hypothesis: *the number of MCSs would have to be less than those of EFMs* [100]. Table 3.2 shows the results obtained with the five networks which are different in size. The columns 2 and 3 remind the size characteristics of these networks. The first given values (in the column 2) are the total number of reactions extracted from the biological descriptions, and the second ones in the parentheses are the number of reactions obtained after computing *enzyme subsets* (see Chapter 1, Section 1.6). The column 3 discloses the number of the internal metabolites.

Table 3.2: The size characteristics and the computation of EFMs and MCSs in the five studied networks. The column 2 consists of two quantities: the number of reactions before and after computing enzyme subsets respectively. The column 3 contains the number of internal metabolites. The last columns contain the computing results for EFMs and MCSs.

Networks	Nb. React.	Nb. Int. Meta.	Nb. EFMs	Nb. MCSs	Avg. (min/max) length of EFMs	Avg. (min/max) size of MCSs
TCA cycle	15(9)	13	16	54	8.3 (4/12)	3.8 (3/4)
Muscle	37(26)	31	3,253	42,534	17.7 (2/23)	10.2 (6/12)
Liver	44 (28)	36	2,307	47,203	16.7 (2/24)	11.4 (6/14)
MNHPC	78 (50)	28	114,614	93,009	37.7 (2/53)	11.1 (4/18)
Aracell	92 (43)	49	1,720,563	43,534	31.8 (1/46)	10.3 (6/12)

➤ **Results**

✓ The first line gives the result for the TCA cycle, which is a part of the mitochondrion metabolism. In fact, it is a single pathway rather than a complex network, but we have chosen

it to mention as an introductory example with the aim of giving an easier explanation of MCSs analysis (see the next section). As we can see in Table 3.2, the number of MCSs is higher than those of EFMs, however, because of these small sets, EFMs and MCSs, we cannot contest anything more about the hypothesis.

✓ The two next lines show the results for the two mitochondrial tissues (Muscle and Liver). These networks are more complete and they describe roughly the energetic metabolisms of a cell. Unfortunately, we can observe that the number of MCSs is **over** 10 **times higher** than those of EFMs.

✓ The two last lines give the results of the plant cell networks, which are larger and more complex than the other ones. At this level of complexity, we **reach the expected behaviour**: less MCSs than EFMs. It is interesting to note that even MNHPC has less reactions than Aracell network, after computing the set of reactions which occurs together (*enzyme subsets*), Aracell network has finally less number of nodes. If we take a look in the EFMs and MCSs results, one can see that the huge results of EFMs for Aracell network (computing with 43 reactions) is not related to the ones of MCSs. This puts emphasis on the fact that EFMs and MCSs do not relay the same information and reveal a different point of view about the network connectivity.

✓ In addition, we have computed the average (and min/max) EFMs length and MCSs size. One can observe that the size of MCSs does not grow up with the size of the networks. Regarding to our results, the number of MCSs seems to have started a going down trend when the network size is increasing, therefore, MCSs approach could be a good candidate to analyse large-scale networks.

3.4 Collaboration between EFMs and MCSs analysis

On the one hand, each EFM is unique and minimal. It implies that no EFM can be a straight composition of some another ones. On the other hand, a not so big network (i.e. containing several tens of reactions) can produce a huge number of EFMs (several hundreds of thousands). Consequently, we can think that almost of them share a lot of similar segments, also called motifs. These motifs can belong to many groups, as overlapped clustering). As the results, most classical clustering algorithms have been failed to apply into the classification of EFMs [135][11].

Finding motifs through the set of EFMs is a way to analyse functional links between the reactions. In order to simply illustrate this purpose, we go back to the production of external

[11] Some works have been done to experiment overlapped clustering to find common motif on EFMs [134]. Unfortunately, from certain size of EFMs sets, the results are so huge that the analysis of them is difficult to handle.

citrate in the TCA cycle network (see Section 1.5.4). Table 3.3 shows the 7 EFMs concerning T1, the reaction that produces citrate. The yellow cells mark the presence of the reactions in the corresponding EFMs. Although it has a less of EFMs, the map shows the hardness of grouping them.

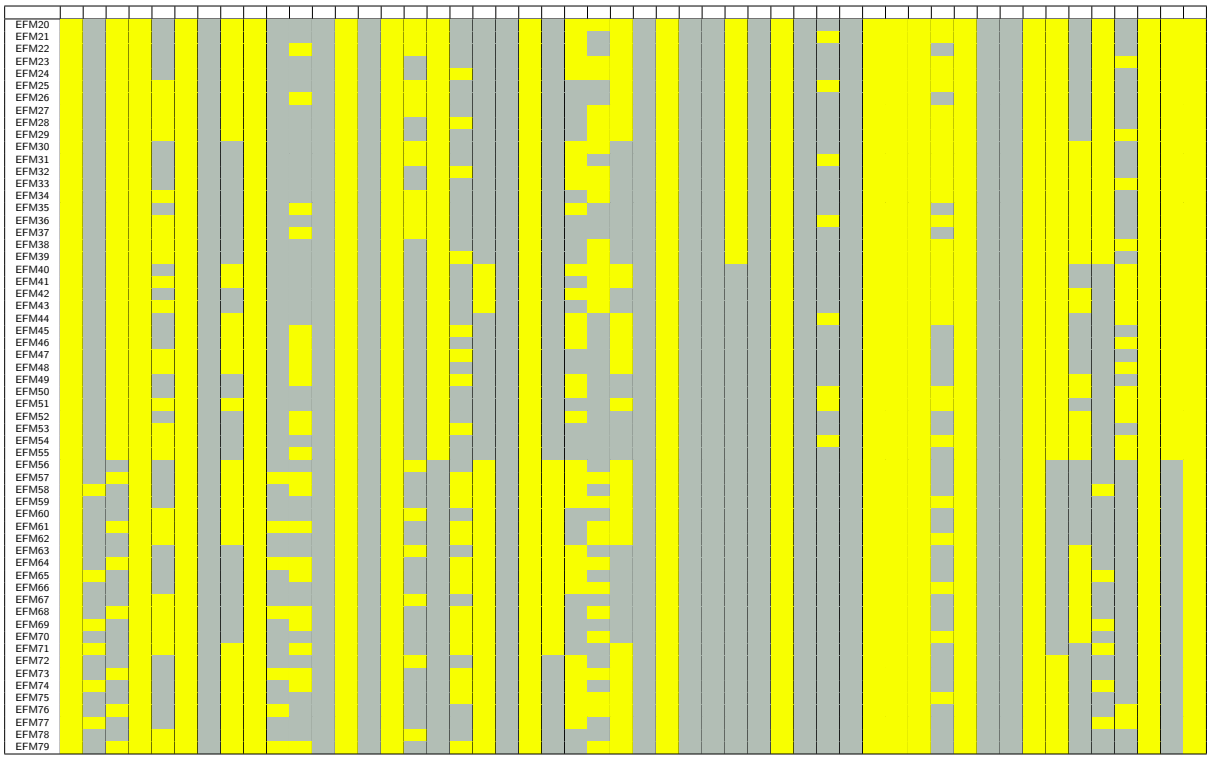
Table 3.3: The yellow colour boxes show the common reactions among 7 EFMs that run T1 in the TCA cycle.

	R6i	R7i	R8i	R9	R10i	R11i	R12	R13	R14	R15	T1	T2	T5	T6	T7	T12
EFM1																
EFM2																
EFM3																
EFM4																
EFM5																
EFM6																
EFM7																

Another more complex example is given in Table 3.4. By selecting 60 EFMs of MNHPC, we have drawn the related map of EFMs and the reactions. The columns and rows correspond to reactions and EFMs respectively. One can see that EFMs share widely same reactions and some do not. We can also remark the variability of EFMs size.

Remark: The snapshot of this map (i.e. the figure) could be concerned the term of file system fragmentation [38], which refers to the condition of a disk in which files are divided into pieces *scattered* around the disk.

Table 3.4: Representation of the complexity in the classification of EFMs in MNHPC



3.4.1 Stating the principal idea

From this point, we have developed the idea to combine EFMs and MCSs results to extract information about relationship between reactions. Based on the initial definitions of EFMs and MCSs, we can deduce the following rules:

➤ We call $|REACTS|$ be the set of reactions in a given metabolic networks with $r = |REACTS|$ and the objective reaction $obR \in REACTS$.

➤ Let $EFM = [Re_1, Re_2, ..., Re_q]$ be a non-decomposable set of the reactions concerning to the objective reaction obR with $q \leq r$ and $Re_q \in REACTS$.

➤ Let $MCS = [Rm_1, Rm_2, ..., Rm_p]$ be a set of the reactions with $p \leq r$ and $Rm_p \in REACTS$. This MCS is one of the optimal solutions stopping reaching to the reaction obR, e.g. preventing the feasible pathway EFM as defined above.

➤ We denote $A \overset{P}{\nrightarrow} B$ to say that *"A blocks the production of P via the path B"*. Hence one can state formally MCS concept following its definition:

$$MCS \overset{obR}{\nrightarrow} EFM \tag{3.4.1}$$

or we can rewrite the above formula:

$$[Rm_1, Rm_2, ..., Rm_p] \overset{obR}{\nrightarrow} [Re_1, Re_2, ..., Re_q] \tag{3.4.2}$$

➤ Consequently, at least one of the reactions in the MCS must be in the EFM so that it makes the EFM being inactive (see the example in Section 3.4.2). Thus, this condition can be presented formally as follows:

$$\exists m_i, e_j (1 \leq m_i \leq p, 1 \leq e_i \leq q) : Rm_i = Re_i \tag{3.4.3}$$

The next section shows a concrete simple example of using these rules.

3.4.2 Stopping the production of external citrate in Krebs cycle

If we go back to the list of the 7 EFMs in the TCA cycle (Figure 3.5), it is possible to extract information from the MCSs list about which reactions could be mandatory.

Computing MCSs of the 7 EFMs provides the list of sets of reactions which are able to stop `citrate` production, i.e. each MCS can disable all EFMs concerning `T1`. In other words, 14 MCSs are considered as the solution that cutting all the pathways to produce `citrate`. Figure 3.6 gives the list of these MCSs.

```
EFM1: R9   R10i T1   T2
EFM2: R9   R10i R11i R12   R13   R14 T1 T5
EFM3: R6i R9    R10i R11i R12   R13 R14 R15 T1 T6 T7
EFM4: R6i R7i   R8i R9 R10i R11i R12 R13 R14 T1 T6 T7
EFM5: R7i R8i   R15 T1 T6
EFM6: R6i R7i   R8i T1 T5 T6   T7
EFM7: R6i R7i   R8i R11i R12 R13   R14   T1 T2   T6   T7
```

Figure 3.5: List of 7 EFMs concerning the production of external citrate

```
MCS1:   T1
MCS2:   R7i    R9
MCS3:   R9     T6
MCS4:   R7i    R11i   T2
MCS5:   R11i   T2     T6
MCS6:   T2     T5     T6
MCS7:   R6i    R9     R15
MCS8:   R6i    R11i   R15    T2
MCS9:   R9     R11i   R15    T5
MCS10: R6i    R7i    T2     T5
MCS11: R6i    R15    T2     T5
MCS12: R7i    R15    T2     T5
MCS13: R9     R15    T2     T5
MCS14: R11i   R15    T2     T5
```

Figure 3.6: List of 14 MCSs disconnect 7 EFMs that ensuring the production of external citrate

Remark: Again, it is worth to note that the number of MCSs (14) is not smaller than those of EFMs (7) but the analysis of MCSs can be considered more easily thanks to their shorter sizes.

Intuitively, [T1] is a trivial minimal cut set in TCA cycle network. To explain more deeply how to interpret MCSs in the context of EFMs, we consider two MCSs (named MCS2 and MCS11) in Figure 3.6. MCS2 (red colour) consists of two reactions R7i and R9. The reactions R9 appears in EFM{1,2,3,4}, whereas the reaction R7i takes part in EFM{4,5,6,7}. Only EFM4 contains both R7i and R9. This can be verified in the same way for the MCS11 (blue colour). Generally, at least one of the reactions from each MCS belongs to each EFM (see Equation (3.4.3)). In our example, each of 7 EFMs contains at least one reaction in red and one in blue. Thus, we can conclude that at least one of the reactions belong to MCS2 and MCS11 and indeed to all MCSs has to present in all EFMs. These reactions constitute motifs that we can observe in EFMs. This property will be used to analyse MNHPC network. The full analysis will be presented in the next chapter and we will see that we have been able to extract a set of core reactions which are groups of controller reactions to produce metabolite of interest.

3.5 Conclusion

It can be said that MCSs, together with EFMs, forms a dual representation of metabolic networks: the MCSs blocking a certain set of target flux vectors are *minimal hitting sets* of the set of EFMs [73, 99]. This idea adds to the increasing importance of Metabolic Pathway Analysis (MPA) and provides a promising tool of finding suitable targets for repressing undesirable metabolic functions, which can be employed in the process of drug target identification [72, 100].

The aim of this chapter was to discuss about minimum cuts in graph theory as well as the concepts of MCSs applying in metabolic networks. As far as we know, no team has worked with traditional algorithms in graph theory for metabolic networks (see Appendix C.3). Then, we have presented the algorithm for computing all MCSs in a metabolic network and its improvements. The tools used in our work have been examined. We have also studied the 5 networks on different structural complexity levels. In one hand, the number of EFMs and MCSs on these networks are computed. The findings reveal that the number of MCSs is higher than the number of EFMs on TCA cycle and mitochondria networks, when the network is not so big and the number of EFMs is not huge, but the number of MCSs is lower than those of EFMs with the bigger networks like the metabolic network of heterotrophic plant cells. On the other hand, the length of MCSs does not increase with the number of reactions, e.g. it becomes stable when the network size grows up. Furthermore, we have also given an example helping to understand the dual relationship between the computation of EFMs and MCSs on TCA cycle network with the stopping the production of external `citrate`. Consequently, MCSs analysis could be an "easy way" to analyse the results of EFMs. And last but not least, the collaboration between EFMs results and MCSs analysis has been discussed. Because of the exponential explosion of the number of EFMs, the results in Section 3.3 arises a question whether or not we can mine MCSs and use them for analysing feasible metabolic pathways. We have shown that MCSs can be used to determine sets of reactions which are jointly mandatory helping to find motifs sharing by EFMs dedicated to a particular function. It has existed few of works taking into account smallest MCSs like us. To the best of our knowledge, The teams working on the MCSs concepts study mainly the improvement of algorithms to compute them and not on how to combine them with another computing. Furthermore, the examples that they provide do not compute all MCSs for a complete set of EFMs. Chapter 4 will present the results we computed on Metabolic Network of Heterotrophic Plant Cells (MNHPC) with the combination of EFMs and MCSs results.

The results obtained in this chapter was:

- published in the article [123] and presented at the conference of advanced in Systems and Synthetic Biology (aSSB) in 2013.

- presented at Metabolic Pathway Analysis at ISGSB 2012 [113] and Metabolic Pathway Analysis 2013 Conference [122].

Chapter 4

Application to Heterotrophic Plant Cell Networks

A bacterium is far more complex than any inanimate system known to man. There is not a laboratory in the world which can compete with the biochemical activity of the smallest living organism.

James Gray
— The Science of Life

In this chapter, we shall describe the full analysis that have been carried out to find the **core** reactions of the **heterotrophic plant cell network**. In order to understand different features and behaviours of this network, we have investigated reactions leading to accumulated metabolites such as **sugars, starch, amino acids**, and **organics acid**. Indeed, these metabolites could be considered one of the main features of fruit metabolism.

4.1 Metabolic Network of Heterotrophic Plant Cells

➤ **Go back to the first step of modelling:** studying particular metabolic network requires getting a list of reactions that form a coherent network. Even the organism genome is published, this task is hard to do because no automatic procedure exists to extract a list of proteins/enzymes from a genome annotation. A lot of research teams are focus on the development of a framework to drive the reconstruction of a specific metabolic network from huge quantity of genomics data but at this time, we observe that a part of this task has to do manually and with the help of an expert of the organism.

We have driven our work taking into account this situation and we have proposed an analysis at the level of a kind of *"middle size"* metabolism. The bottom level of a metabolic analysis is to model one or several enzymatic reactions which belong to a pathway and drive one biological function such as the *glycolysis* or the *TCA* cycle. Mainly this level is covered by models like

differential equations which are able to describe most often clearly the evolution of concentration of one flux but fail to model interactions between several pathways. As we have briefly presented in Chapter 1, Flux Balance Analysis (FBA) is a way to model interaction of several pathways through a network and to obtain quantitative measures of phenomena. Many researchers demonstrate renewed attention for this method because currently performing machines allow to measure a lot of metabolites in one experiment. However, the lacking of many information as kinetics parameters or exact behaviours of enzymes limit the interpretation of obtained results. In contrast, at the top level of metabolism analysis, whole reconstruction of a metabolic network could be achieved from genomics information, but on the one hand, organism genome are not all available, on the other hand as we have said, no automatic procedure is available at this time.

Our purpose has been to explore networks at the middle size levels/scales from analysing a network with several pathways but focus on a subset of the whole metabolism. For example, we have mainly analysed networks without taking into account genetic regulation or signalling. The first reason of this choice is that we have been able to obtain consistent description at this level of metabolic networks from our collaboration with biologists and the second, we think that it is useful to provide such an analysis to complement quantitative analysis [1] and to provide a deep insight about collaborations and/or competitions between enzymes through the network. Now we shall pay attention to our main application of MNHPC.

4.1.1 Description of the first version of the network

First, we look at the model of our metabolic network model, Metabolic Network of Heterotrophic Plant Cells (MNHPC) (Figure 4.1), given by our collaborators in LaBRI[1] and INRA[2] team. This network includes the main pathways of the central carbon metabolism in plants: glycolysis (black), the TCA cycle (blue), the pentose phosphate pathway (pink), the starch and sucrose pathways (green) and the storage reactions towards the vacuole (brown). Due to its autotrophic nature, the plant synthesises its own respiratory substrates (mainly carbohydrates) which then serve as substrates for the TCA cycle. The TCA cycle provides precursors for several biosynthetic processes, such as nitrogen fixation and biosynthesis of amino acids [107]. The pentose phosphate pathway includes the irreversible oxidative branch, whereas the non-oxidative branch is reversible (recycling of pentose-phosphates from fructose phosphate and triose-phosphate). In the starch and sucrose pathways, sucrose is metabolized in cytosol, whereas starch is metabolized in plastids from imported hexose phosphates (`G1P` or `G6P`). Several effluxes are illustrated: protein synthesis from several amino acids (glutamate and glutamine, aspartate and alanine), lipid synthesis (diacyl glycerol) from plastidial pyruvate and trioses, synthesis of cell wall polysaccharides from UDP-glucose, sugars (glucose, fructose and sucrose) and storage of organic acids

[1] http://www.labri.fr
[2] http://www.bordeaux-aquitaine.inra.fr/

(malate and citrate) in vacuoles. Subcellular compartments, such as mitochondria and plastids, can lead to potentially reversible transport of metabolites such as `G6P`, `X5P`, `PEP` and `DHAP`. This network has been published in a previous paper [21].

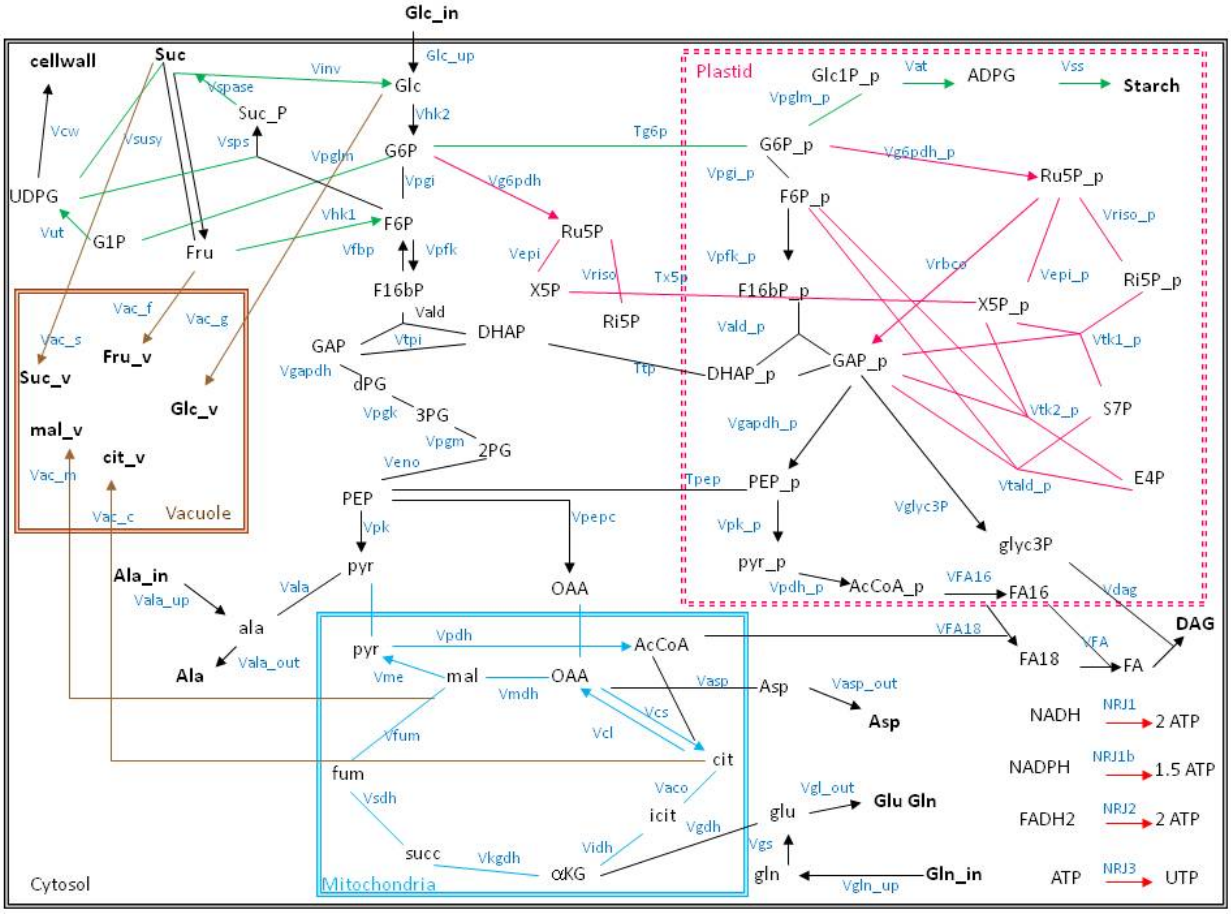


Figure 4.1: Metabolic network of a heterotrophic plant cells. Each colour indicates one pathway: blue for the TCA cycle, black for glycolysis and also for the fluxes towards output metabolites, pink for the PPP, green for the sucrose and starch synthesis, red for respiration and brown for storage in vacuole. External metabolites are in bold. Irreversible reactions are indicated by unidirectional arrows.

The description of MNHPC contains 70 different metabolites and 78 reactions including 15 external metabolites and 33 reversible reactions. The external metabolites are carbon sources or carbon sinks (nutrients, waste products, stored and excreted products, and precursors for further transformation). These are exogenous glucose and amino acids (glutamine and alanine), CO2, sugars (sucrose, glucose and fructose) and organic acids (citrate and malate) stored in vacuoles, amino acids for protein synthesis (aspartate, alanine, glutamate and glutamine), cell wall polysaccharides, starch and lipids. The metabolites named cofactors (ATP, NADH, NADPH) are internal ones which means that they are balanceable at steady state. The full description file of MNHPC is given in Appendix A.4. This file is in METATOOL format [154].

4.1.2 Description of the redefined network

In order to analyse MNHPC, described above, we have used the software CellNetAnalyzer (CNA) as the initialised tool to find feasible pathways. The first step of the computing procedure is to find sets of reactions which always operate together in feasible pathways within steady state of the system, called *enzyme subsets* [136][3].

➤ List of subset composition

We have found 12 subsets of reactions which concern in the 38 reactions. Table 4.1 shows the list of these subsets. The column **New equation** resumes for each subset the new substrates, products conversion. To be more readable, we have given a name for each subset using a composition of their own reaction names.

Table 4.1: List of the sets of reactions/enzymes (also called enzyme subsets) replacing in MNHPC. The new reaction names are the associations of the old ones because of its simpleness and usability

No.	Set of enzymes	New equation
1	Vgapdh, Vpgk, Vpgm, Veno	GAP $\Longleftrightarrow$ PEP + ATP + NADH
2	Vaco, Vidh	cit $\Longleftrightarrow$ aKG + NADH + CO2
3	Vriso_p, Vtkx_p, Vtald_p	Ru5P_p + 2 X5P_p $\Longleftrightarrow$ 2 F6P_p + DHAP_p
4	gln_up, Vgs	aKG + NADPH + gln_in $\Longrightarrow$ 2 glu
5	Vg6pdh, Vepi, Tx5p	G6P $\Longrightarrow$ X5P_p + NADPH + CO2
6	Vpfk_p, Vald_p, Vtpi_p	F6P_p + ATP $\Longrightarrow$ 2 DHAP_p
7	Vpk_p, Vpdh_p, VFA, VFA16, VFA18, Vdag, Vglyc3P	4 AccoA + 3 DHAP_p + 48 PEP_p + 4 ATP + 88 NADPH $\Longrightarrow$ 45 NADH + 3 DAG + 48 CO2
8	Vat, Vss, Vpglm_p	G6P_p + ATP $\Longrightarrow$ starch
9	Vut, NRJ3, Vpglm	G6P + ATP $\Longrightarrow$ UDPG
10	Vsps, Vspace	F6P + UDPG $\Longrightarrow$ Suc
11	NRJ2, Vkgdh, Vsdh, Vfum	aKG $\Longrightarrow$ mal + 2 ATP + NADH + CO2
12	Vasp, Vasp_out	OAA + glu $\Longrightarrow$ aKG + asp_out

The majority of these subsets are not surprising because they are series of linear reactions. For example, the subset [`Vgapdh`, `Vpgk`, `Vpgm`, `Veno`] includes the reactions occurring continuously in pathways (see Figure 4.1). The case of the subset [`Vg6pdh`, `Vepi`, `Tx5p`] has a little bit different because it exists a branch with the reaction `Vriso`. At the reaction `Vg6pdh`, we have two branches to pass the processing: (1) to continue with `Vepi` and `Tx5p` (2) to go through `Vriso`. After the first step of analysis of the stoichiometric matrix, no feasible pathway could be built using `Vriso`. Thus, the reaction `Vriso` can be removed out of the list of the candidate reactions without changing the network behaviours. It is worth to note that by the way we are enable to allow biologists to verify pathway schemes of reconstructed metabolisms.

[3] An enzyme subset consists of several reactions expressed simultaneously in a given metabolic pathway.

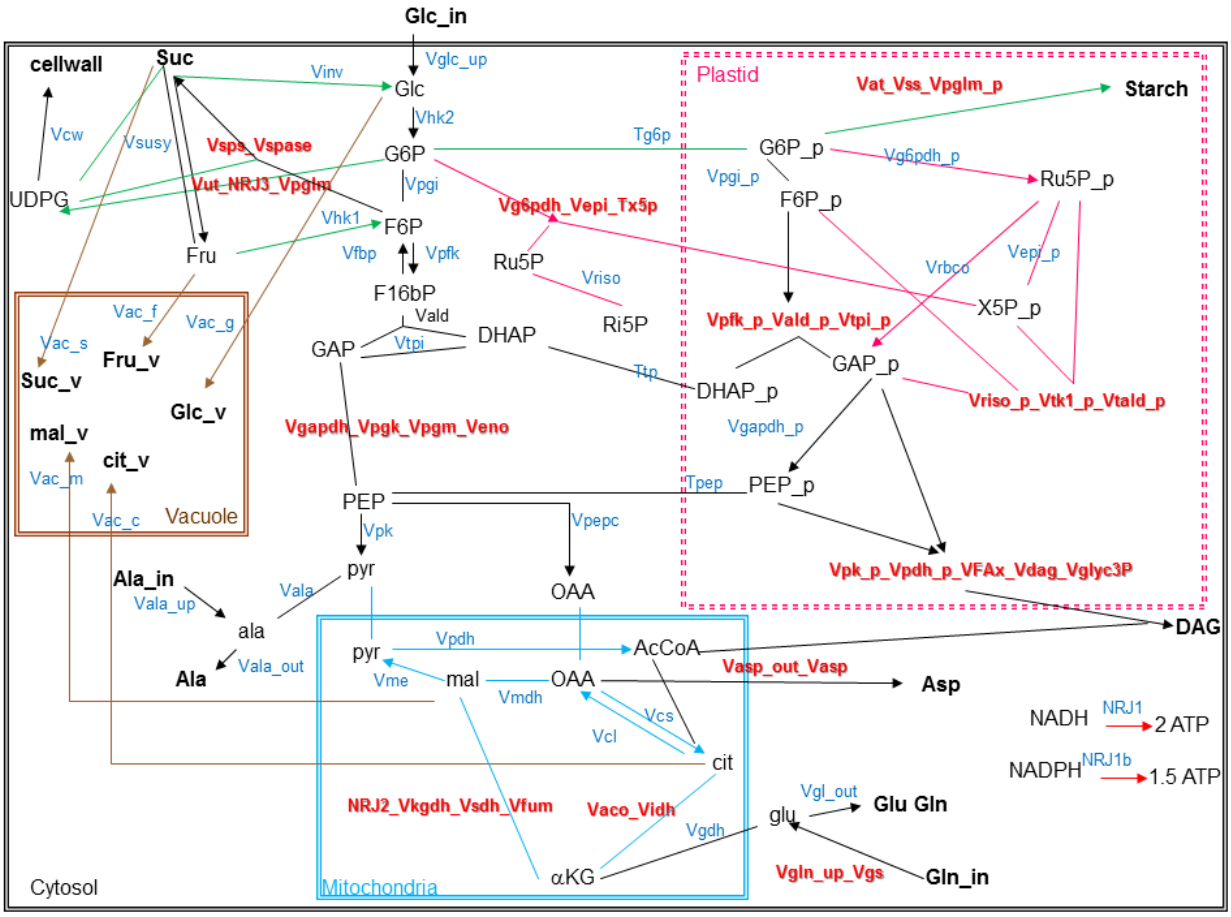


Figure 4.2: Enlarged redefined metabolic network of a heterotrophic plant cells. Each colour indicates one pathway: blue for the TCA cycle, black for glycolysis and also for the fluxes towards output metabolites, pink for the PPP, green for the sucrose and starch synthesis, red for respiration and brown for storage in vacuole. External metabolites are in bold. Irreversible reactions are indicated by unidirectional arrows. The new reactions replace the their enzymes subsets to be depicted in shadow, bold and red colours.

Prior to computing EFMs and MCSs, the data file of MNHPC is rewritten by replacing the old reactions with the new ones as shown in Table 4.1. The final network version has 43 metabolites and 49 reactions including 15 external metabolites and 14 reversible reactions. Table 4.2 resumes the differences in size between the original and new description after computing enzyme subsets.

Table 4.2: Differences in size between the original and redefined version of MNHPC

	Nb. Reactions		Nb. Metabolites	
	Reversible	**Irreversible**	**Internal**	**External**
The original network	33	45	55	15
The redefined network	14	35	28	15

Remark: From this point, all the analyses have been done on the redefined version of MNHPC unless otherwise is specified.

The following paragraph shows the result of computing global structural properties that is considered the basic analysis of MNHPC.

4.1.3 Computation of global structural properties

As mentioned in Chapter 2, our metabolic network MNHPC can be modelled by a directed graph. The network modelling of MNHPC consists of 92 vertices (nodes) and 149 edges (arcs). The set of nodes consists of two types: *metabolite nodes* and *reaction nodes*.

Using the graph extraction method presented in Section 2.1.3, we have built reaction and metabolite networks based on MNHPC. The reaction network composes of 49 vertices and 261 edges, whereas the metabolite network has 43 vertices and 131 edges (as summarised in Table 4.3). To analyse MNHPC in more details, we shall work with these two networks and compare them to the complete one in the following sections.

Table 4.3: Topological properties of three networks: MNHPC, reaction network and metabolite network.

	MNHPC	Reaction network	Metabolite network
Number of vertices	92	49	43
Number of edges	149	261	131

As we have addressed in Chapter 2, degree distribution can be computed for all networks, no big differences have been found between the values obtained for the first version of MNHPC and the reduced one.

Moreover, we have also noticed that MNHPC has only one *connected component*. That means it always exists at least a pathway connected from a node to all nodes in MNHPC. This strong connection approves the close coordination between the elements inside the network.

✓ To compute and manage all the data we have presented in this PhD thesis, we have sometimes used existing tools, and it is worth to note that MNHPC has been used in collaboration with regEfmtool [88]. We have often written a lot of pieces of code to build pipeline between tools, to use existing algorithms for graph cuts or to implement our own algorithms. For example, all these programs have not been assembled into a framework at this time. One of the reasons is the fact that they are heterogeneous (C++, Python, MATLAB languages) but the expertise that we have gained during this work lead us to plan to finalise such platform as soon it will be possible.

4.1.4 Computation of Elementary Flux Modes

As it was mentioned in Chapter 1, CellNetAnalyzer (CNA) [102] and regEfmtool [88] are not only two available tools for computing EFMs. At the beginning of this work, we have used the software CNA which provides a friendly graphics interface. Unfortunately, the running and computing time are extremely expensive with large-scale networks. Fortunately, regEfmtool runs more times faster than CNA. To benchmark the computing performances, the algorithms have been tested in the same configuration of the computer. For this purpose, we have used a Linux server 64 bits Intel(R) Xeon(R) CPU X5675 3.07GHz consisting of 24 cores 1.6GHz, cache size 12MB and 94GB RAM. For MNHPC, CNA run the batch to compute EFMs in more than 15 continuous days without any interruption while regEfmtool is able to extract the $114,614$ EFMs in 12s.

In our opinion, the only drawback of CNA is the implementation of algorithms in MATLAB environment, while regEfmtool is extended from the open source version of *Efmtool* java program which is more efficient because of the new data structure, *bit tree*, to store the matrix and more speed programming language. Consequently, we have preferred to use regEfmtool mainly for our study.

➤ Occurrences of reactions

Several researches discussed about hubs [6, 17] inside networks to find essential metabolites or reactions. To take a look if such hubs reactions can be found from the EFMs set, we have computed the occurrences of each reaction in the set.

The histogram in Figure 4.3 shows the occurrence of reactions participated in EFMs.

The first group contains 9 reactions which are present less than 20% of EFMs. The second, third and fourth groups contain reactions which are present between 25% and 85% of EFMs. These three groups are equivalent in size. Finally, 3 reactions belonging to the last group could be considered as essential because they participate in more than 95% of EFMs. These are 3 reactions `Glc_up`, `Vhk2` and `NRJ1`. The first two ones concern the main entry of glucose and the last one is an energy reaction. Thus, no surprising information can be found. We can only notice `NRJ2` which is the other energy reaction does not belong to this group, very probably because now it is included in a subset and linked to 3 another ones (i.e. its using is constraint). At the opposite, the group of the less used reactions are mainly the output reactions. The histogram suggests that a core of more than 30 reactions is mainly used in all solutions with different combinations.

➤ Length of EFMs

As we do not take into account the kinetics of reactions, it is impossible to argue that a short EFM is faster than a long one. But the length can give us an information about the complexity

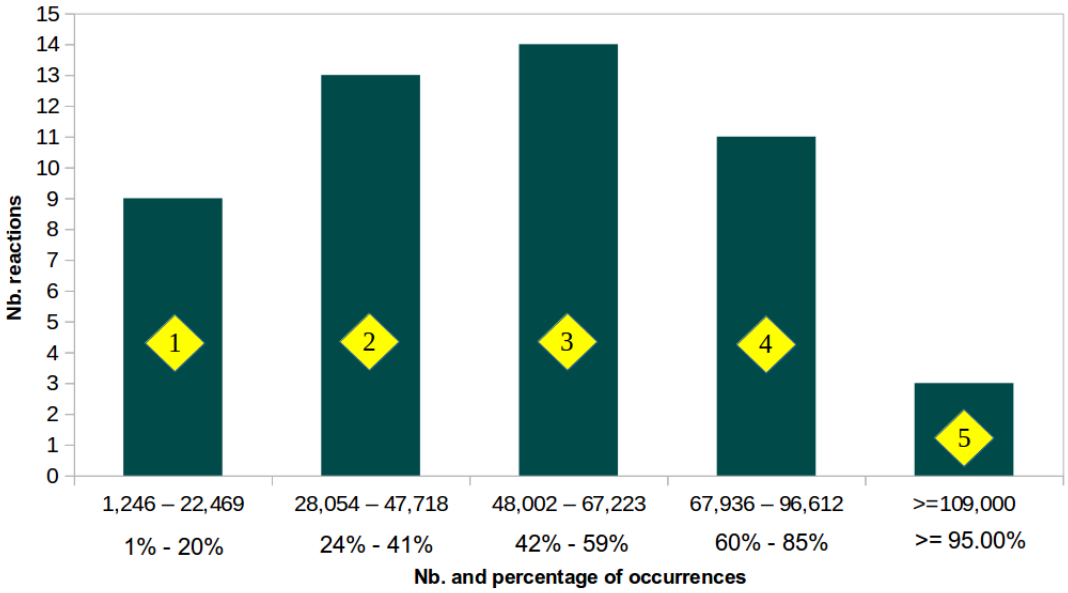


Figure 4.3: Histogram of the occurrences of the reactions in the set of EFMs of MNHPC.

to obtain some metabolites. Figure 4.4 shows the histogram of EFMs length which fluctuates between 19 and 28.

It is clear that the number of EFMs rises dramatically from the length 2 to the length 24 and then goes down at the length 25 until the end at 28. However, the number of EFMs with length 2 to 17 are inconsiderable. For instance, there are 2 EFMs the length 2 (e.g. [ala_up,Vala_out]; [Glc_up,Vac_g]) that are pathways playing the role of exchangeable input/output metabolites. There exists exactly one EFM with length 5 (e.g. Vgapdh_p, Vtpi, Ttp, Tpep, Vgapdh_Vpgk_Vpgm_Veno) and the following table gives the very small effective of EFMs for "smallest length". In contrast, the length 28 nearly reaches at $2,500$ EFMs (approx. 2%).

Length	10	11	12	13	14	15	16	17
Nb. EFMs	7	6	17	49	124	302	728	1,373

Almost all of EFMs have the length from 18 to 27. In other words, there are more than $10,900$ EFMs (approx. 95%) which length belongs to the range from 18 to 27. In fact, MNHPC has 49 reactions and nearly 50% of them participates in feasible metabolic pathways. As the result, it shows the plasticity of the given network.

➤ Classification of EFMs

Classification is one of the methods for exploring data complexity. We have tried to distribute EFMs into smaller clusters by using ACOM algorithm [135]. One can note that ACOM was

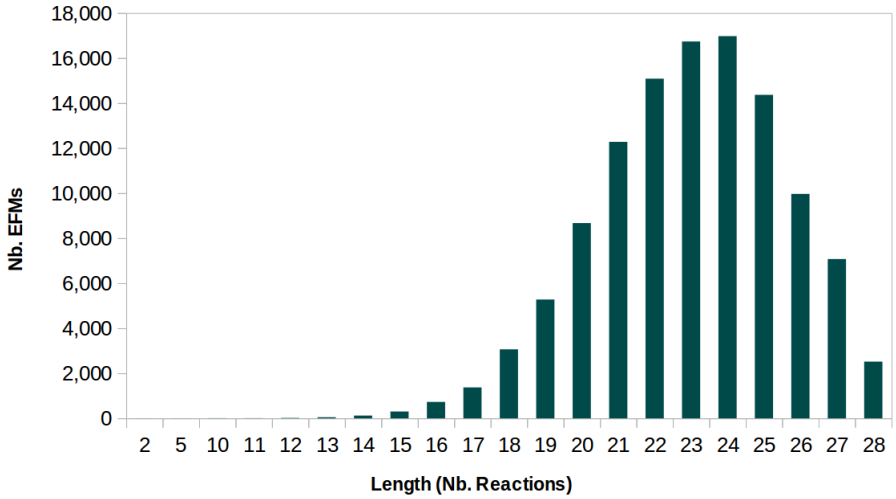


Figure 4.4: Histogram of the pathway lengths of the EFMs in the global network MNHPC

developed by Perès et al. [132] and tested on three mitochondria which network sizes are smaller than MNHPC. Unfortunately, some problems appear with ACOM running to MNHPC and the program generates a number of clusters exceeding our expectations, about several hundreds of classes. The amount of classes do not allow doing efficient classification of EFMs.

With $114,614$ EFMs and the analyses above, seeking feasible pathways and interpreting biological issues meaningfully is remarkably difficult. In the following paragraphs, we shall reach the next step: computing the dual solutions of EFMs, that is minimal cut sets.

4.1.5 Computation of Minimal Cut Sets

As mentioned in Section 3.2, MCSs computation can be performed with CNA or mcsCalculator [85]. The server served for this computing is the one as described and used in EFMs computation (see Section 4.1.4). For MNHPC in the same task of computing the whole of MCSs, CNA runs more than 15 continuous days while mcsCalculator needs more than 15 minutes to finish the same batch, therefore, the results that we present here are obtained from mcsCalculator[4].

Seeing the result of MNHPC as discussed in Section 3.3, we have found $93,009$ MCSs. It is clear to note that the number of MCSs is smaller than those of EFMs but already not possible to analyse manually.

➤ Occurrences of reactions

Normally, the first step of studying a large set of MCSs is to compute the occurrence/frequency of reactions participated in MCSs with the purpose of finding which part of the network we have to focus on.

[4]mcsCalculator is available only from the middle of 2013 and we have benefited of it from this time.

Figure 4.5 shows the histogram about the occurrence of reactions in MCSs. The data are divided into 6 groups based on the percentage of the occurrences of the reactions. It is noticeable that the number of the occurrences varies considerably.

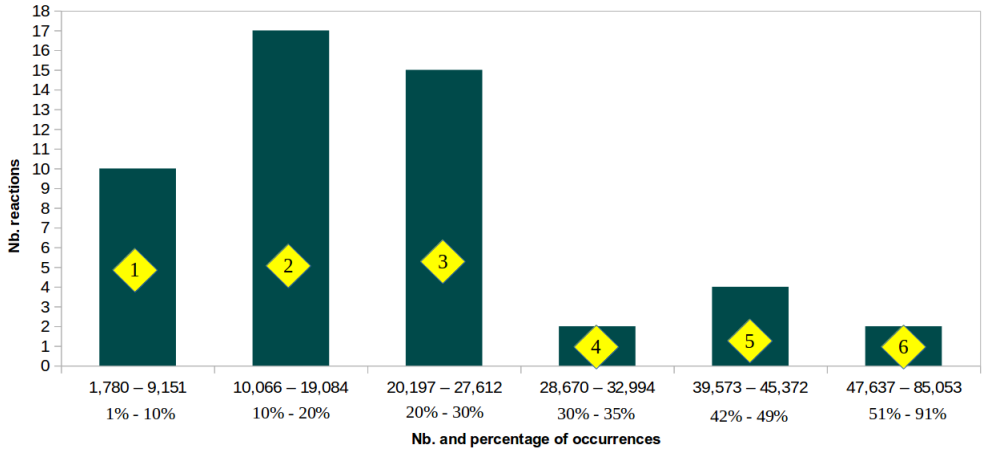


Figure 4.5: Histogram of the occurrence of reactions in the set of MCSs in MNHPC

We can see that most of the reactions occur sparsely in MCSs. Indeed, the three first groups (e.g. the length $1, 2$, and 3 as depicted Figure 4.5) contain almost all of the reactions (e.g. 42 reactions) with the proportional occurrence less than 30% generally.

It is clear that the occurring frequencies of reactions in MCSs are not denser than the case of EFMs as sketched in Figure 4.3. The percentage of the most occurrences of reactions in EFMs spread wide from 1% until roughly 84% whereas the similar measured values of MCSs are between 1% and under 30%. Henceforth, we can imagine that the size of interrelated reactions will be smaller and easier to analyse than those of EFMs.

➤ Size of MCSs

The histogram in Figure 4.6 shows the size of MCSs varying between 4 and 18. That confirms some analyses that we have shown in Chapter 3.

There are 28 MCSs with the smallest length 4 and 30 MCSs have the longest length 18. The greatest value $17,347$ MCSs belongs to the length 11. It is noticeable that the length of MCSs is more stable than EFMs. In other words, MCSs length is independent on network size.

➤ Classification of MCSs

We have also tried classifying the set of MCSs with ACOM algorithm. The results obtained are not really feasible because the input parameters supplied to ACOM are unstable. Thus, we have decided to analyse MCSs in another way.

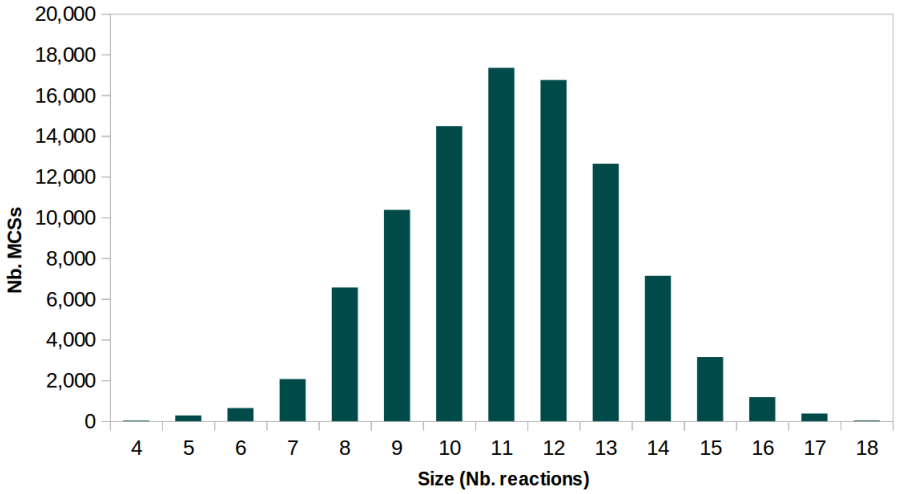


Figure 4.6: Histogram of the size of MCSs in MNHPC

4.2 Analysis of specific metabolic productions

At this step of the analysis of MNHPC, we have two large sets of results and the next goal is to use them to point out behaviours of the network.
Biologists can easily point out a list of metabolites that they want to outline. A list of 5 metabolites have been constituted: **starch**, **fructose**, **glucose**, **sucrose** and **glutamate** because the measurement of the production of these metabolites are considered as the main parameters to assess the plant growth and development.
The question formulated by biologists is: *"how we can produce these metabolites if the entry of glucose, the reaction `Glc_up`, is stopped?"*.

4.2.1 The reason of choosing five cases

In order to understand plant growth and to know how to improve the production and the quality of their products. Experiments using **glucose** are generally performed because they have much relevant biological information and are easy to implement.

We have studied the production of fructose and glucose in the `Vacuole` compartment corresponding to the two reactions `Vac_f` and `Vac_g` in MNHPC, respectively. Both of the reactions `Vac_f`, `Vac_g` charge of the production of common monosaccharides sugars (named as glucose, galactose and fructose). Meanwhile, the other reaction `Vac_s` located in the same compartment takes into producing sucrose (one of three common disaccharides sugars). `Vgl_out`, which belongs to the `Cytosol` compartment, is the reaction occurring at the end point of the replacement of `Glc` (Glucose) with `Gln` (Glutamine). The other metabolite in our study located in the `Plastid` compartment is `starch` which can be found in large amounts in fruits, seeds, rhizomes, and tubers, as well as photosynthetic tissues [65]. `Starch` molecules are polymers of

glucose. Vss is the last reaction in the chain of reactions that produces starch metabolite.

After the identification of the reactions to focus on, the next step is to present how they are structured.

4.2.2 Presentation of the five sub networks

From the complete network, the question is risen that how are the five sub networks corresponding to the five given metabolites built?

Actually, from the global matrix of EFMs containing all feasible pathways of MNHPC, we have extracted all EFMs containing each of the reactions Vac_f, Vac_g, Vac_s, Vgl_out and Vss and formed the groups of EFMs regarding to the appropriate target reactions. The five EFMs matrices have been built and each of them represents one of these reactions. As it has been explained, an EFM is a list of reactions. So from each matrix, it is possible to extract the list of reactions which are implied to the production of each metabolite of interest and to know which one is absent. This operation of re-modelling of the network guarantees to work correcting thanking to the definition of EFMs.

To be more convenient, the reaction names will be used to identify each sub network in explanations, figures and tables as well. For example, using the sub network Vac_g is dedicated to run the reaction Vac_g.

In order to apply classical algorithms of graph theory, it could be interested to design the graph corresponding to each EFMs matrix. From the list of reactions in the original description of MNHPC, it is possible to extract a list of nodes and a list of edges involving each matrix. All the reactions nodes with the number of occurrences equals to zero is eliminated and the corresponding edges too. Now, we present five subgraphs of the original one, containing all the reactions and metabolites implied on each production.

We can see in Table 4.4 that the networks Vss and Vac_s do not use the reaction Vac_g and have the same number of reactions although they are located inside two different compartments. The network Vac_g has 6 reactions not use while the other sub networks basically maintain in stable (e.g. the number of missing reactions is not remarkable as given in Table 4.4). The network Vac_g has different behaviours from the others with the missing 6 reactions. It is worth to note that the reaction Vac_g appears in the sub networks Vac_f, Vgl_out (i.e. they are able to produce glucose in Vacuole) but missing in the cases Vss, Vac_s.

➤ Sub networks without the uptake of Glc_up

In MNHPC, the main entrance of glucose is modelled by the reaction Glc_up. The missing Glc_up will affect other functions because it is one of the reactions occurring most in EFMs with

Table 4.4: List of the reactions to be missed after extracting specific metabolites of interest. The sub networks mentioned are built from the complete network MNHPC without any mention of missing `Glc_up` or not. The leftmost column signifies the reactions that are missing in the network.

Missing reactions	Network				
	Vss	Vac_s	Vac_f	Vac_g	Vgl_out
Vpfk_p_Vald_p_Vtpi_p				x	
Vhk2				x	
Vat_Vss_Vpglm_p				x	
Vcw				x	
Vac_s				x	
Vpfk				x	
Vac_g	x	x			
Nb. Reactions	48	48	49	43	49
Nb. Nodes	79	85	81	85	86
Nb. EFMs	22,469	19,392	34,752	1,246	19,608

the highest frequency as computed and interpreted in Section 4.1.4. To response the question of what happens if `Glc_up` is missing, we have extracted from each matrix the corresponding one with EFMs not containing `Glc_up`.

Table 4.5 shows the number of reactions remaining and the names of the reactions not used in the 5 sub networks. All the 5 networks reduces the number of reactions participating in metabolic pathways where the most amount is over 25% (the network `Vss` has 13 reactions to be unused) and the least amount equals 12% (the network `Vgl_out` has 6 reactions to be unused).

Moreover, we can observe in Table 4.5 that `Vac_s` and `Vss`, both of them have the same 10 unused reactions. These sub networks are pretty the same. In other words, these two metabolic processes in our plant cell model could have many similarities. `Vac_f` and `Vac_g` have the same the number of reactions but they have one difference at existing the reaction `Vhk1` and `Vhk2` respectively.

In conclusion, the modifications via the removal of the unused reactions make some parts of the networks inactive. To see all 5 models of the sub networks after removing the unused reactions, we attached the drawings corresponding to the 5 sub networks in Appendix D.2.

Now, we shall move in the following paragraphs to address the effects on the sub networks due to the missing of `Glc_up`.

4.3 Effects of stopping the entrance of glucose

As mentioned above, the missing of glucose affects the production of the objective metabolites of the five sub networks. Thus, we shall present some measures about this influence in order

Table 4.5: List of the unused reactions of 5 sub networks if `Glc_up` is stopped.

Missing reactions	Network				
	Vss	Vac_s	Vac_f	Vac_g	Vgl_out
Vpfk_p_Vald_p_Vtpi_p	x	x	x	x	
Vpfk	x	x	x	x	
Vhk1	x	x	x		
Vhk2	x	x		x	
Glc_up	x	x	x	x	x
Vcw	x	x	x	x	
Vsusy	x	x			
Vac_c					x
Vac_f	x	x			
Vac_g	x	x			
Vac_m					x
Vac_s	x		x	x	
Vinv	x	x			
Vsps_Vspace	x				
Vut_NRJ3_Vpglm	x				
Vat_Vss_Vpglm_p		x	x	x	
Vpk_p_Vpdh_p_VFAx_Vdag_Vglyc3P					x
NRJ1b					x
NRJ2_Vkgdh_Vsdh_Vfum					x
Nb. Reactions	37	39	43	43	44
Nb. Nodes	70	74	81	81	82
Nb. Nodes with `Glc_up`	79	81	85	85	86
Nb. EFMs without `Glc_up`	415	415	833	1,245	754
Nb. EFMs with `Glc_up`	22,469	19,392	34,752	1,246	19,608

to find similarities and differences among these sub networks.

4.3.1 Connectivity of the sub networks

To check the networks which have just remodelled, we have verified parameters as diameter, characteristic path length or coefficient clustering. All the obtained values for the five sub networks are quite similar to the global MNHPC network, even for the complete, reaction or metabolite networks (built following the rules explained in Chapter 2). Since our purpose does not focus on these measures, we have not presented them in details. But we have concluded that no more useful information could be retained with these parameters in our case.

To order to figure out which elements belong to the networks are essential, we have continued to find reactions and metabolite hubs.

4.3.2 Reaction hubs and metabolite hubs

The next properties of the sub networks that we have studied are centers and peripheries. The center is the set of nodes with eccentricity equal to the radius whereas the periphery is the set of nodes with eccentricity equal to the diameter. It is worth to note that metabolic networks rely

heavily on a few crucial metabolic hubs, such as ATP, NADH, and CO2, that are well-known to be used widely in many cellular biochemical reactions. In contrast to hubs, most other metabolites each participate in only a few reactions as they were often stated [110, 175] .

Table 4.6: Centers and peripheries of 5 sub networks

	center	periphery
Vss	ATP, CO2, NADH, NADPH, PEP, *NRJ2_Vkgdh_Vsdh_Vfum, Vaco_Vidh*, *Vme*, *Vpdh*	ala_in, ala_out, *Vfbp*
Vac_f	ATP	ala_in, ala_out, Fru_v, gl_out
Vac_g	ATP	ala_in, ala_out, Glc_v, gl_out
Vac_s	ATP	ala_in, ala_out, gl_out, Suc_v
Vgl_out	ATP	ala_in, ala_out, gl_out, Suc_v

➤ Centrality and eccentricity of the five sub networks can be seen in Table 4.6. Obviously, the metabolites ATP and the others such as CO2, PEP, NADH, NADPH etc. produced much and take part in most of biological processes are main substances playing the role of centers. The reactions NRJ2, Vaco, Vidh, Vfum, Vsdh, Vkgdh, Vme, Vpdh used frequently in many metabolic pathways are centers.

4.3.3 The occurrences of reactions and the length of EFMs

Definitely, $114,614$ EFMs of MNHPC consist of ones containing and not containing the reaction Glc_up. In the case of missing the entrance of glucose, i.e. the reaction Glc_up is inactive, obviously EFMs containing Glc_up will not work. Thus, to analyse the effect of stopping the entrance of glucose, obviously we have chosen working with the set of EFMs not containing the reaction Glc_up. Table 4.7 shows the statistics summary about the two sets of EFMs containing or not containing Glc_up for the five studied networks.

➤ Comparison of the occurrences of reactions

✓ As we have known, Glc_up is the main reaction assuring the entrance of glucose. So the number of EFMs with or without Glc_up in the columns 2 and 4 in Table 4.7 reveals the differences between two sets in size. It seems that the datasets in the case of without Glc_up reach a size which can be considered as manageable. So we begin with a list of easy observations.

✓ Vac_g has unique EFMs with the size 2 containing Glc_up that is the entrance of metabolism. So this feasible pathway mainly starts from Glc_up and ends at Vac_g that they are primary

Table 4.7: Comparison of the number of EFMs and their lengths of the sub networks with MNHPC.

Network	EFMs with Glc_up			EFMs without Glc_up		
	Nb. EFMs	Avg. size	Min/Max	Nb. EFMs	Avg. size	Min/Max
MNHPC	109,224	22.97	2/28	5,390	22.97	2/28
Vss	22,054	22.73	12/28	415	21.04	13/24
Vac_f	33,919	24.07	14/28	833	25.00	13/28
Vac_g	1	2.00	2/2	1,245	24.71	16/28
Vac_s	18,977	23.66	14/28	415	23.04	15/26
Vgl_out	18,854	22.36	14/28	754	22.59	13/27

reactions in forming the pathway. It is a trivial EFM.

✓ The number of EFMs containing `Glc_up` in `Vac_s` and `Vgl_out` is close.

✓ The columns 3 and 5 in Table 4.7 show the length of EFMs in both cases. The lengths vary from 12 to 28 for both cases except for the network `Vac_g`.

➤ Comparison of the histogram of the EFMs lengths

In addition, we have tried comparing the histogram of the complete network MNHPC in Figure 4.4 and the histograms of the networks missing of `Glc_up` in Figure 4.7. The aim is to see whether the missing of `Glc_up` influences on the reactions taking part in EFMs or not.

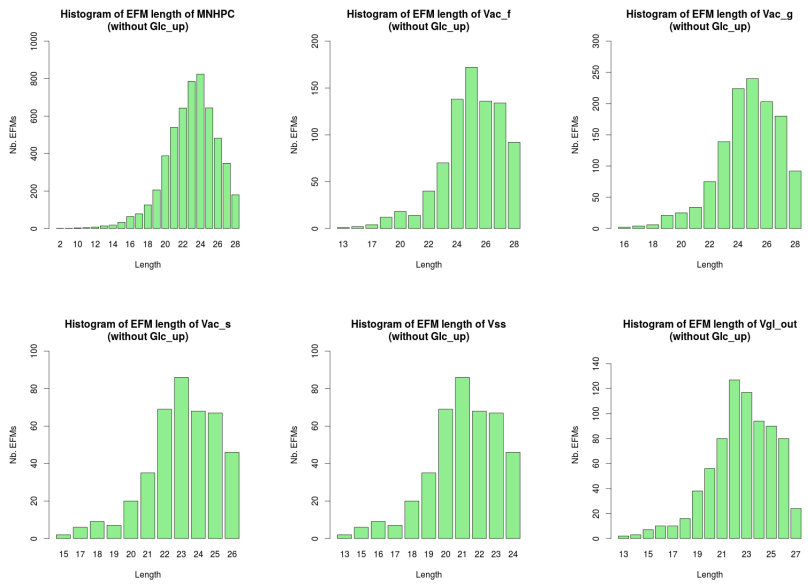


Figure 4.7: Histogram of EFM length of MNHPC and the five sub networks.

✓ The histogram of the EFM lengths of all the networks reveals that the average length does not change noticeably in Figure 4.7. The values at the peaks equal to the average lengths of all the networks and it is true if comparing to the average length of EFMs of the complete network MNHPC as shown in Figure 4.4. It confirms that the length of EFMs in most of the cases is not affected by changing the network behaviours.

4.3.4 Combining MCSs result and EFMs analysis

In our study, we have inspected the data and calculated EFMs as well as MCSs for seeking some interesting views. We shall present the results of the combination of MCSs computation with EFMs analysis and connect to the relevant explanations.

➤ Comparison of the number of EFMs and MCSs

From the relevant matrices, we have computed MCSs which are able to stop the production of five interest metabolites. Table 4.8 reminds the total number of EFMs for each sub network in the column 2. The column 3 contains their number of MCSs. From that result, we can extract all MCSs containing `Glc_up` as shown in the column 4.

Table 4.8: Comparison of the number and the length of EFMs and MCSs.

Network	Nb. EFMs	Nb. MCSs	Nb. MCSs with `Glc_up`
Vss	$22,469$	$13,901$	15
Vac_f	$34,752$	$14,446$	15
Vac_g	$1,246$	562	561
Vac_s	$19,392$	$14,473$	15
Vgl_out	$19,608$	$5,500$	87

✓ In the network `Vac_f`, the number of EFMs is round 2.5 times greater than the one of MCSs while the number of MCSs in the network `Vgl_out` is about 3 times smaller than the one of EFMs. The number of EFMs in the network `Vac_g` is roughly twice bigger than the one of MCSs.

✓ Numerically speaking, however, the networks `Vss`, `Vac_f` and `Vac_s` exhibit the same number of MCSs, and comparing their contents could be easy. At the opposite, almost all MCSs of `Vac_g` include `Glc_up` reaction. The network `Vgl_out` has a middle result, even there are more numerous than the smaller group, the result seems to be handleable.

➤ Comparison of the histogram of MCS sizes

It is worth to compare the histogram of MCSs size of the networks containing `Glc_up` to the one of MNHPC (Figure 4.6). Generally, the total number of MCSs has decreased a lot, and so

the histogram of the size has not the same shape.

✓ Figure 4.8 contains the histograms of the size of the five networks. The histograms of the three networks `Vss`, `Vac_f`, `Vac_s`) are similar in the shape: they contain only MCSs of size 2 and 3. The three other ones have more MCSs, therefore, the histograms of size are different. The histograms reveal that a part of the amount of MCSs are small. This result seems to confirm that these MCSs could be analysed easily.

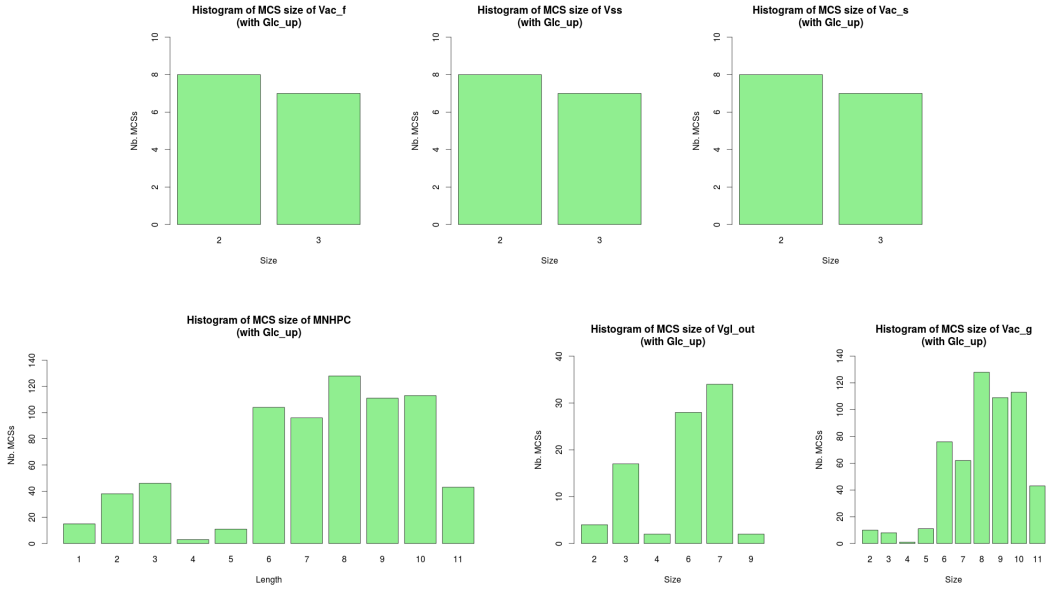


Figure 4.8: Histogram of the MCS size of MNHPC and the five sub networks. We are addressing the networks which MCSs contain `Glc_up`.

➤ Taking into account smallest MCSs

For all the sub networks, the number of MCSs as well as their average lengths are always smaller than the corresponding aspects of EFMs. This observation can be enable to confirm that using MCSs seems to be easier for studying the set of results. One way is to analyse the smallest MCSs such as the ones of size $2, 3, 4$, etc. with the aim of understanding which sets of reactions have functional links. Using smallest MCSs to analyse the network has been pointed out for a few months by Kamp and Klamt [176] who have proposed a method for the effective analysis of these MCSs. Our results achieved in this study have supported to find back common motifs in EFMs that shall be presented in the next sections.

➤ Finding "core" reactions using MCSs of size 2

First, we can extract a list of MCSs of size 2 containing `Glc_up` for each matrix. Second, it is essential to remind that if `Glc_up` is stopped, the other reaction belonging to a MCS of size 2 is mandatorily preserved by definition of MCSs (see Equation (3.4.3)). Finally, collecting

all these reactions from any MCSs of size 2 builds a list of reactions which are mandatory to produce the five metabolites of interest. As a result, we have found 8 reactions **Vpgi**, **Vfbp**, **Vpgi_p**, **Vrbco**, **Tg6p**, **Vald**, **Vriso_p** and **Vepi_p** which occur mandatorily in EFMs (red colour in Figure 4.9). The group of these eight reactions can be considered as the **"core"** of our MNHPC for the production of the metabolites of interest that we have selected.

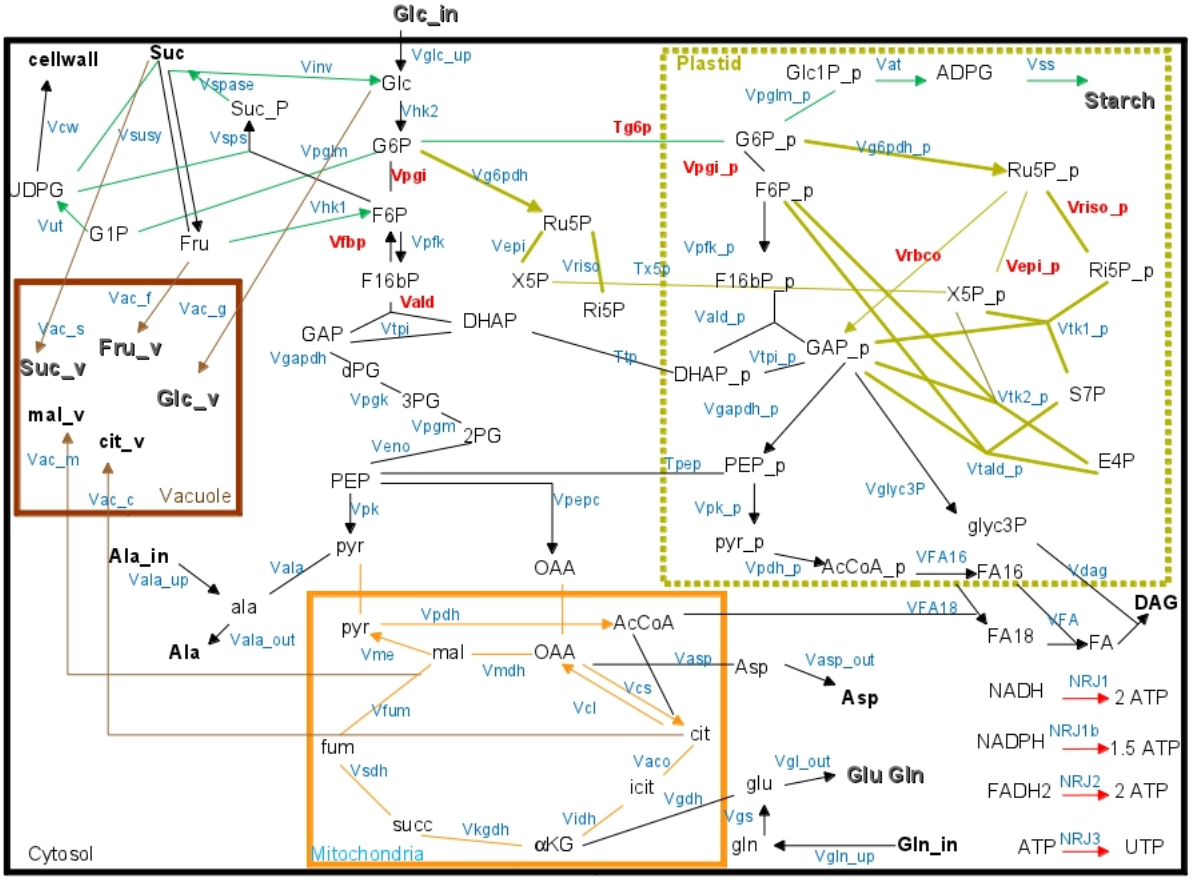


Figure 4.9: Enlarged metabolic network of a heterotrophic plant cells with 8 mandatory reactions highlighted. The interpretation of the other colours is similar to the explanations in Figure 4.2.

➤ Branching in EFMs

Identification of reactions which belong to all EFMs corresponding to a specific function as the production of sugar or amino acids is not really difficult and can be obtained by several ways. But for the next step, finding set of reactions which are relevant to group EFMs together in an efficient classification, is less easy. We have already mentioned that both generic clustering methods and specific one (i.e. we have developed previously) are failed to solve this problem when the number of EFMs is huge.

The method that we proposed is to continue to use smallest MCSs. In this step, we are taking into account the MCSs of size 3 to identify a new set of reactions which are almost mandatory.

➤ Finding branches using MCSs of size 3

The same procedure employed with MCSs of size 2 has been applied to MCSs of size 3: collecting the two other reactions belonging to these MCSs in association with `Glc_up`. In the case of MCSs of size 3, we obtain now a list of couple of reactions, and the rule is *"at least one of the two reactions has to be included in the EFMs"*. In other words, we can find two branches, one for each reactions. In summary, MCSs of size 3 provide a list of **"branching points"** (i.e. reactions) in the metabolic network.

Table 4.9: List of all MCSs of size 3 containing `Glc_up` of the five networks. `Vgapdh*Veno` is short for Vgapdh_Vpgk_Vpgm_Veno.

Vss			**Vac_f**		
Vgapdh_p	Vtpi	Glc_up	Vgapdh_p	Vtpi	Glc_up
Vgapdh*Veno	Vtpi	Glc_up	Vgapdh*Veno	Vtpi	Glc_up
Ttp	Vtpi	Glc_up	Ttp	Vtpi	Glc_up
Ttp	Vgapdh_p	Glc_up	Ttp	Vgapdh_p	Glc_up
Ttp	Vgapdh*Veno	Glc_up	Ttp	Vgapdh*Veno	Glc_up
Tpep	Vtpi	Glc_up	Tpep	Vtpi	Glc_up
Tpep	**Ttp**	Glc_up	Tpep	Ttp	Glc_up
Vac_g			**Vac_s**		
Vsps_Vspace	Vsusy	Glc_up	Vgapdh_p	Vtpi	Glc_up
Vgapdh_p	Vtpi	Glc_up	Vgapdh*Veno	Vtpi	Glc_up
Vgapdh*Veno	Vtpi	Glc_up	Ttp	Vtpi	Glc_up
Ttp	Vtpi	Glc_up	Ttp	Vgapdh_p	Glc_up
Ttp	Vgapdh_p	Glc_up	Ttp	Vgapdh*Veno	Glc_up
Ttp	Vgapdh*Veno	Glc_up	Tpep	Vtpi	Glc_up
Tpep	Vtpi	Glc_up	Tpep	Ttp	Glc_up
Tpep	Ttp	Glc_up			
Vgl_out					
Vepi_p	Vgdh	Glc_up			
Vepi_p	Vg6pdh_p	Glc_up			
Ttp	Vtpi	Glc_up			
Ttp	Vgapdh_p	Glc_up			
Ttp	Vgapdh*Veno	Glc_up			
Tpep	Vtpi	Glc_up			
Tpep	Ttp	Glc_up			
Tg6p	Vgdh	Glc_up			
Tg6p	Vg6pdh_Vepi_Tx5p	Glc_up			
Tg6p	Vepi_p	Glc_up			

4.3.5 Motif branches into MNHPC

➤ List of possible branches

Based on the method has just been discussed, we have collected all MCSs of size 3 containing Glc_up as shown in Table 4.9. Five reactions **Vgapdh_p, Vtpi, Vgapdh_Vpgk_Vpgm_Veno, Ttp** and **Tpep** playing the role of the **"branching points"**, have been found. The number of EFMs containing these branching points has determined in Table 4.10. Of course, these results could include EFMs containing several of the five reactions. Next, we have chosen the case of the branch Tpep/Ttp (values in red in Tables 4.9 and 4.10) to explain these results.

In Table 4.10, the results could be considered as strange because we obtained most the same value for each branch. For example in the network Vss, 311 EFMs contain Tpep and 311 Ttp. The question is *"how many of these EFMs contain both reactions and how many only one of them?"*. The result is 207 EFMs containing both and 104 only Tpep and 104 Ttp. The same computing has been done for all branches and provided some range of results.

➤ Studying association of branches

From Table 4.9, we can see that Ttp can be also branched with Vtpi. Thus, it exists a kind of combinations of branches through EFMs. To be clear, we give an example as follows.

✓**Example of branches in EFMs** In order to understand the branching in EFMs, we show here an introductory illustration: alignment two EFMs belonging to Vac_f to see the similarities and differences between them as given follows. To facilitate the lecture we have suppressed some common reactions and kept the relevant part of the EFMs for this explanation.

```
NRJ2_Vkgdh_Vsdh_Vfum Vhk2 Vfbp Vpdh Vcs Vgapdh_p Vrbco Vinv NRJ1
~~~~~ Vac_f Vac_c Vpgi Vald Vtpi Vmdh Vpgi_p Vepi_p Tg6p Tpep Vala Vgdh
Vgapdh_Vpgk_Vpgm_Veno Vriso_p_Vtkx_p_Vtald_p

NRJ2_Vkgdh_Vsdh_Vfum ~~~~ Vfbp Vpdh Vcs ~~~~~~~~ Vrbco Vinv NRJ1
Vac_g Vac_f Vac_c Vpgi Vald Vtpi Vmdh Vpgi_p Vepi_p Tg6p Ttp  Vala Vgdh
~~~~~~~~~~~~~~~~~~~~ Vriso_p_Vtkx_p_Vtald_p
```

In these two EFMs, we can see several differences in colour. The first one follows an association of Tpep, Vgapdh_Vpgk_Vpgm_Veno and Vgapdh_p branches, the second one includes only the Ttp branch. All the other reactions are the same. This is the main observation that we can do: if we consider EFMs through the *filter of branches*, we can exhibit large part of EFMs which are in common.

Table 4.10: Number of EFMs in which not containing `Glc_up` but having one of 5 given reactions. EFMs of course have the participation of 8 mandatory reactions. `Vgapdh*Veno` is short for `Vgapdh_Vpgk_Vpgm_Veno`.

Network	Total number	Tpep	Ttp	Vtpi	Vgaphd_p	Vgapdh*Veno
Vss	415	311	311	311	311	311
Vac_f	833	624	624	624	624	624
Vac_g	1,245	933	933	933	933	933
Vac_s	415	311	311	311	311	311
Vgl_out	754	559	559	574	559	559

Branching of EFMs can be observed more clearly by visualising all EFMs in graph - a kind of **motif graph**, called **EFMs graph**. The **EFMs graph** is built from a **root** node which **vertices** are reactions. Two nodes are connected together if they belong to the same EFM. All EFMs can be stored in an **EFMs graph** where you could have a look its branches.

For instance, EFMs of the network `Vac_f` creates the branches beginning at the node `Tpep` and `Ttp` visualised as Figure 4.10. At the node `Tg6p` (i.e. one of eight **core** reactions) of the **EFMs graph**, we can follow one of the two branches: one goes with `Tpep` and the other follows `Ttp`. Combinations of `Tpep` and `Ttp` can be happened in order to create another branches where have the presence of these reactions.

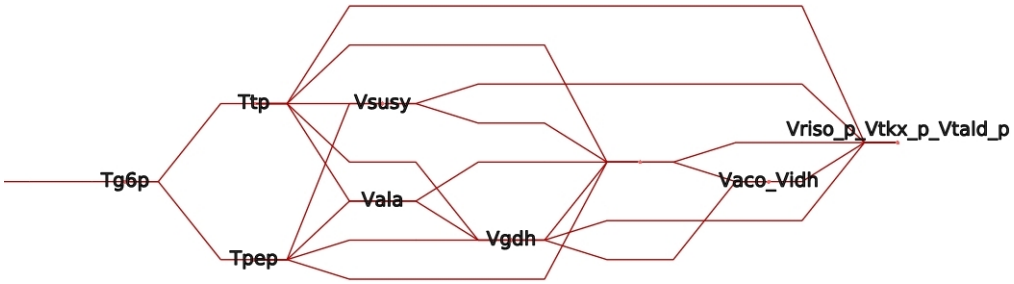


Figure 4.10: Explanation of the branching in EFMs via visualising all EFMs in a tree.

➤ Final comparison of EFMs branches

The last step of our analysis has constituted on identification of group of associated branches. Table 4.11 resumes the results for `Vss`, `Vac_f`, `Vac_g` and `Vac_s` taking into account 10 cases of combinations of branches. `Vgl_out` is not compared in this step because it does not share the same combinations than the other ones (probably because it concerns another part of the network).

The procedure that we have applied is the following:

- First, get in each matrix the set of EFMs containing one branch reaction, for example, `Ttp` and `Veno` subset.

- Second, sets are compared two by two. For each line belonging to each set, we compare the lines and suppress the similarities.
- Finally, we collect the remain reactions in each lines and create a two parts of motifs. The first part is the set of reactions belonging to the first set of EFMs and the second for the other one.

For example, the comparison between EFMs containing `Ttp` and those containing the enzyme subset `Vgapdh_Vpgk_Vpgm_Veno` gives us 26 different combinations (given in the first column of Table 4.11). Appendix D.3 gives the list of these 26 combinations.

We can observe that for the four networks, the result is the same. We can state that the branch combinations obtained are the same for the production of these four metabolites of interest.

Another useful result that we have achieved is the list of combinations of the ten case studies as in Table 4.11. The table shows that the main difference of two branches focuses on some lists of reactions. The lists are supplied in Appendix D.3.

Table 4.11: Different motifs of ten pairs of branches. The different motif consists of the reactions kept after removing the similar reactions.

Network	Ttp-Veno	Tpep-Vtpi	Ttp-Vgap	Tpep-Ttp	Ttp-Vtpi	Vtpi-Veno	Vtpi-Vgap	Veno-Vgap	Tpep-Vgap	Tpep-Veno
Vss	26	29	32	29	1	26	32	29	6	24
Vac_f	26	29	32	29	1	26	32	29	6	24
Vac_g	26	29	32	29	1	26	32	29	6	24
Vac_s	26	29	32	29	1	26	32	29	6	24

The number of these combinations are very small if we compare to the initial set of EFMs that we have taken into account. From Table 4.11, we can conclude that for four metabolites of interest, it exists, for example for `Ttp/Veno` branch, 26 different motifs of reaction combinations that characterise the pathways. These motifs build from the MCSs of size 3 allow us to gather EFMs which exhibit the same behaviour: same list of reactions plus one part of the possible combinations. It signifies that for example if the reaction `Ttp` is stopped for any kind of reasons, the motif `Tpep` + `Veno` subset can replace it with exactly the same other list of reactions, the same input metabolites and the same output.

4.4 Conclusion and Future works

In this chapter, we have analysed our middle size Metabolic Network of Heterotrophic Plant Cells (MNHPC) in order to find the **hub reactions** and the behaviours of the network when some

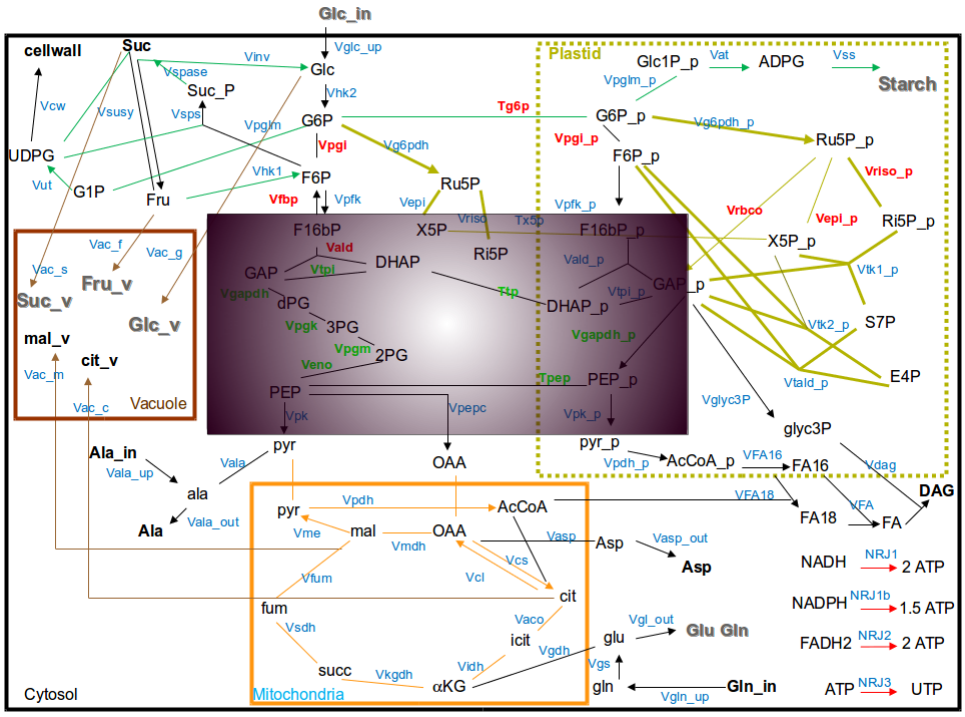


Figure 4.11: Metabolic network of a heterotrophic plant cells and 5 interest reactions. These reactions are in green under the rectangle filled with a gradient.

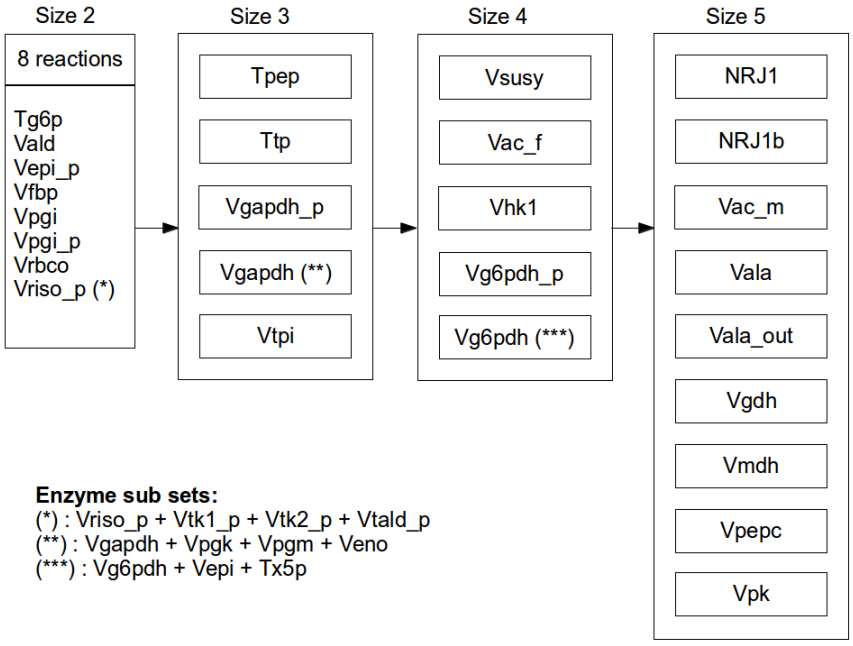


Figure 4.12: MCSs-based model of seeking motifs and branches in huge sets of EFMs. Our model has worked with MCSs of size $2, 3, 4$ and 5.

reactions do not work. To do that, we have built five matrices of feasible pathways, EFMs, dedicated to the production of the five metabolites of interest: **starch**, **fructose**, **glucose**, **sucrose** and **glutamate**. The question is *"what will happen when the main entrance of glucose is stopped?"*. By using small size MCSs, we have set of list of 8 reactions which are mandatory. These reactions are the **core** of the network. In the second step, the MCSs of size 3 have provided a list of 5 reactions which are branches through the network: **branching points**. Figure 4.11 resumes these results and shows that even the 13 reactions stay in the center of the network, their identification was not so obvious. For example, any one from mitochondria compartment belongs to this set.

To end this conclusion, we want to tell that this research was a prospective work about using EFMs and MCSs conjunction. We have had to manage a huge amount of data and spend much time to study them. From our experience and the amount of piece of code that we have written, we are ready to define a useful framework to automate the large part of this work.

➤ Future works

To pursue our work, first we can enlarge and complete the protocol of using MCSs to study EFMs. Currently, we have succeeded in analysing MCSs of size 2 and 3 in the context of MNHPC. The MCSs of size $4, 5$ and 6 in MNHPC are sparse, but the preliminary results (see Figure 4.12) shown that the new list of reactions to take into account is not at all huge.

This research has shown network organisation of heterotrophic plant cell metabolism via studies on specific metabolites/functions. Another task that could be conducted is to expand the list of these functions by adding other metabolites.

To the best of our knowledge, no available tools can visualise well the desired subsets of EFMs as well as MCSs. Using information like branch reactions could be helpful to manage new drawing algorithms.

Conclusion

In the last decade, a lot of biological networks have being built from the large scale experimental data produced by the rapidly developing high-throughput techniques as well as literature and other sources. This not only opens many chances for bioinformaticians but also is one of challenges that experimenters have to face with. Computer scientists have employed computerised tools, especially graph theory, to model such biochemical reaction graphs and to analyse their desired behaviours.

This PhD research was set out to explore topological analysis of metabolic networks, the concepts of EFMs and MCSs and to propose the combination of these methods. Large set of connected metabolic pathways are well-known to be difficult to analyse. Computing the global structural measures belonging to graph theory aims to evaluate the complexity level of our networks and to determine the network structure (like random, small-world or scale-free. . .). This computation could be used to determine whether topological/structural analysis might help us studying network organisation. But from our concrete experiences, with different network sizes: mitochondria muscle and liver and two different versions of plant cell network, these parameters revealed being not really efficient to study network organisation. Consequently, we have turned to compute feasible pathways, the EFMs, and their dual representation, i.e. set of reactions, which are able to stop these feasible pathways, i.e. the MCSs. We have done several measures on the set of EFMs: finding reaction hubs via computing frequency of occurrences of the reactions and comparing the results among different networks; comparing the length of the EFMs and the size of the MCSs. From this first step of analysis, we have stated that even the result size of MCSs remains smaller than EFMs, several thousands of solutions have to be analysed.

In the second step, we have proposed a new way to analyse EFMs with the help of MCSs. The main idea behind the combination of two methods is *"at least one of the reactions belonging to a certain minimal cut set has to be included in all EFMs"*. Thus, if one of these reactions is stopped, the other ones has to be maintained to ensure feasible pathways happening. Apart from the comparable global results for the mitochondria and plant cell networks, the

main application has been performed on the metabolic network of heterotrophic plant cell. At the next step, biologists have selected 5 metabolites of interest: **starch**, **fructose**, **glucose**, **sucrose** and **glutamate** and their production without the help of **glucose** entrance has been studied. We have computed both the set of corresponding matrices of EFMs and the set of MCSs. Using the MCSs of size 2, we have established a list of 8 reactions which are mandatory to produce the 5 metabolites in absence of **glucose**. Using the MCSs of size 3, we have defined a list of 5 reactions which are alternative branches that pathways have to follow. By applying these results to the set of feasible pathways to produce the 5 metabolites of interest, we have identified sets of alternative motifs in pathways which are equal otherwise. Only some tens of motifs of branching points have been identified in our sets of EFMs that reduces a lot the number of cases to analyse.

Structural network analysis attempts to elucidate functional features from the network topology. This type of analysis investigates the general constraints on network behaviour. The scheme proposed by combination of identification of feasible routes and ways to cut these routes, provides a filter that can be used to know alternative solutions to produce metabolites of interest. One of the main problems at this time is that community of biologists wants to reach the level of the cell/organism network to analyse and to experiment functioning of metabolic network. But a list of several tens or hundred reactions produces a level of complexity which is totally different than the one they are accustomed. The resulting graph of reactions is never possible to understand just by looking it. The generic tools coming from graph theory are difficult to reuse in this context. The specific tools as EFMs and MCSs are not really popular in the community of biologists and one of the main reasons is that it misses concrete procedures and interest applications to use them. We have found a way to characterise a metabolic network from a list of mandatory reactions, the core of the network, and a list of branching points, to control a set of objective reactions. As the future work, we can propose a framework to automate this procedure and as the further perspective to reuse this information to support automatic drawing algorithms for metabolic networks which are mainly still missing.

Bibliography

[1] Abdou-Arbi, O., Lemosquet, S., Van Milgen, J., Siegel, A., and Bourdon, J. (2014). Exploring metabolism flexibility in complex organisms through quantitative study of precursor sets for system outputs. *BMC systems biology*, 8(1):8. [13, 64]

[2] Abe, H. and Go, N. (1981). Noninteracting local-structure model of folding and unfolding transition in globular proteins. II. Application to two-dimensional lattice proteins. *Biopolymers*, 20(5):1013–31. [130]

[3] Abel, U. and Bicker, R. (1982). Determination of All Minimal Cut-Sets between a Vertex Pair in an Undirected Graph. *IEEE Transactions On Reliability*, R-31(2):167–171. [46, 141]

[4] Acuna, V., Chierichetti, F., Lacroix, V., Marchetti-Spaccamela, A., Sagot, M.-F., and Stougi, L. (2009). Modes and cuts in metabolic networks: Complexity and algorithms. *Biosystems*, 95(1):51–60. [18, 52, 53]

[5] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. [142]

[6] Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *World Wide Web Internet And Web Information Systems*, 74:47–97. [32, 69]

[7] Albert, R., Jeong, H., and Barabasi, A. (2000). Error and attack tolerance of complex networks. *Nature*, 406(6794):378–82. [13, 36]

[8] Amaral, L. A. N., Scala, A., Barthélémy, M., Stanley, H. E., and Barthelemy, M. (2000). Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(21):11149–11152. [32]

[9] Amitai, G., Shemesh, A., Sitbon, E., Shklar, M., Netanely, D., Venger, I., and Pietrokovski, S. (2004). Network analysis of protein structures identifies functional residues. *Journal of molecular biology*, 344(4):1135–46. [35]

[10] Anthonisse, J. M. (1971). The Rush In A Directed Graph. CWI Technical Report Stichting Mathematisch Centrum. Mathematische Besliskunde-BN 9/71, Stichting Mathematisch Centrum - Mathematische Besliskunde. [38]

[11] Ariyoshi, H. (1972). Cut-set graph and systematic generation of separating sets. *IEEE Transactions on Circuit Theory*, 19(3):233–240. [45, 141]

[12] Arunkumar, S. and Lee, S. H. (1979). Enumeration of All Minimal Cut-Sets for a Node Pair in a Graph. *IEEE Transactions On Reliability*, R-28(1):51–55. [45, 141]

[13] Balcioglu, A. (2000). An algorithm for unumerating the near-minimum weigt s-t cuts of a graph. Master's thesis, NAVAL POSTGRADUATE SCHOOL, Monterey, California. [45]

[14] Ballerstein, K., von Kamp, A., Klamt, S., and Haus, U.-U. U. (2012). Minimal cut sets in a metabolic network are elementary modes in a dual network. *Bioinformatics*, 28(3):381–387. [53]

[15] Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. page 11. [32, 33]

[16] Barabási, A.-L. and Bonabeau, E. (2003). Scale-free Networks. *Scientific American*, 288:60–69. [33]

[17] Barabási, A.-L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113. [13, 18, 69]

[18] Barrat, A. and Weigt, M. (2000). On the properties of small-world network models. *The European Physical Journal B - Condensed Matter and Complex Systems*, 13(3):547–560. [32]

[19] Berg, J. M., Tymoczko, J. L., and Stryer, L. (2002). *Biochemistry*, volume New York. W. H. Freeman and Company, San Francisco, 5 edition. [5, 6]

[20] Berge, C. (1987). *Hypergraphs: combinatorics of finite sets*. Elsevier Science. [127, 128]

[21] Beurton-Aimar, M., Beauvoit, B., Monier, A., Vallée, F., Dieuaide-Noubhani, M., and Colombié, S. (2011). Comparison between elementary flux modes analysis and 13C-metabolic fluxes measured in bacterial and plant cells. *BMC Systems Biology*, 5(95). [65]

[22] Beurton-Aimar, M., Parisey, N., Vallée, F., and Colombié, S. (2010). Identification of functional hubs through metabolic networks. In *ALifeXII,12th International Conference on the Synthesis and Simulation of Living Systems*. Odense. [18]

[23] Billinton, R. and Allan, R. N. (1992). *Reliability Evaluation of Engineering Systems: Concepts and Techniques*, chapter 11, pages 347–350. Kluwer Academic/Plenum Publishers, New York, 2 edition. [43, 142]

[24] Bollobás, B. (1986). *Combinatorics: set systems, hypergraphs, families of vectors and combinatorial probability*. Cambridge University Press, Cambridge. [127]

[25] Bordbar, A., Monk, J. M., King, Z. A., and Palsson, B. O. (2014). Constraint-based models predict metabolic and associated cellular functions. *Nature reviews. Genetics*, 15(2):107–20. [12]

[26] Bornholdt, S. and Schuster, H. G. (2003). *Handbook of Graphs and Networks: From the Genome to the Internet*. John Wiley & Sons, Inc. [25]

[27] Botafogo, R. A. (1993). Cluster Analysis for Hypertext Systems. In *Proc. of the 16th Annual ACM SIGIR Conference of Res. and Dev. in Info. Retrieval*, pages 116–125. [43]

[28] Boykov, Y. and Jolly, M.-p. (2000). Interactive organ segmentation using graph cuts. In *In Medical Image Computing and Computer-Assisted Intervention*, pages 276–286. [143]

[29] Burgard, A. P., Pharkya, P., and Maranas, C. D. (2003). Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and bioengineering*, 84(6):647–57. [52]

[30] Chaouiya, C. (2007). Petri net modelling of biological networks. *Briefings in Bioinformatics*, 8(4):210–219. [15, 129]

[31] Chatterjee, S., Gilbert, J. R., Schreiber, R., and Shefl:ler, T. J. (1994). Array distribution in data-parallel programs. In *In Proceedings of the Seventh Workshop on Languages and Compilers for Parallel Computing*, pages 76–91. Springer-Verlag. [43]

[32] Clarke, B. L. (1980). *Stability of Complex Reaction Networks*, chapter 1, pages 1–215. John Wiley & Sons, Inc. [14, 46]

[33] Clarke, B. L. (1988). Stoichiometric network analysis. *Cell biophysics*, 12:237–253. [14, 46]

[34] Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-Law Distributions in Empirical Data. *SIAM Review*, 51(4):661–703. [33]

[35] Crucitti, P., Latora, V., Marchiori, M., and Rapisarda, A. (2003). Efficiency of Scale-Free Networks: Error and Attack Tolerance. *Physica A: Statistical Mechanics and its Applications*, 320:622–642. [35]

[36] Curet, N. D., DeVinney, J., and Gaston, M. E. (2002). An efficient network flow code for finding all minimum cost s-t cutsets. *Computers & Operations Research*, 29(3):205–219. [137]

[37] de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology : a journal of computational molecular cell biology*, 9(1):67–103. [13]

[38] de Nijs, G., Biesheuvel, A., Denissen, A., and Lambert, N. (2006). The effects of filesystem fragmentation. In *Proceedings of the Linux Symposium*, volume 1. Citeseer. [57]

[39] del Sol, A. and O'Meara, P. (2005). Small-world network approach to identify key residues in protein–protein interaction. *Proteins: Structure, Function, and Bioinformatics*, 58(3):672–682. [35]

[40] Deo, N. (1974). *Graph Theory with Applications to Engineering and Computer Science*. Automatic Computation. Prentice-Hall, Inc. [24, 44, 141]

[41] Eiter, T. and Gottlob, G. (1995). Identifying the Minimal Transversals of a Hypergraph and Related Problems. *SIAM Journal on Computing*, 24(6):1278–1304. [127]

[42] Elias, P., Feinstein, A., and Shannon, C. (1956). A note on the maximum flow through a network. *IRE Transactions on Information Theory*, 2(4):117–119. [132]

[43] Emmert-Streib, F. (2011). A Brief Introduction to Complex Networks and Their Analysis. In Dehmer, M., editor, *Structural Analysis of Complex Networks*, chapter 1, pages 1–26. Springer. [130, 131]

[44] Erdös, P. and Rényi, A. (1959). On random graphs. *Publicationes Mathematicae*, 6:290–297. [131]

[45] Erdös, P. and Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61. [131]

[46] Eriksson, A. P. and Barr, O. (2006). Image Segmentation Using Minimal Graph Cuts. [46, 143]

[47] Estrada, E. and Rodriguez-Velazquez, J. A. (2005). Complex Networks as Hypergraphs. *Physica A: Statistical Mechanics and its Applications*, 364:581–594. [129]

[48] Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 251–262. ACM. [33]

[49] Faust, K., Dupont, P., Callut, J., and van Helden, J. (2010). Pathway discovery in metabolic networks by subgraph extraction. *Bioinformatics*, 26(9):1211–1218. [35, 39]

[50] Fell, D. A. and Wagner, A. (2000). The small world of metabolism. *Nature Biotechology*, 18:1121–1122. [13, 18, 35, 41]

[51] Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181. [143]

[52] Folger, O., Jerby, L., Frezza, C., Gottlieb, E., Ruppin, E., and Shlomi, T. (2011). Predicting selective drug targets in cancer through metabolic networks. *Molecular systems biology*, 7:501. [9]

[53] Ford, L. R. and Fulkerson, D. R. (1956). Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8:399–404. [132]

[54] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5):75–174. [39]

[55] Frank, A. (1994). On the edge-connectivity algorithm of Nagamochi and Ibaraki. [138]

[56] Fredman, M. L. and Khachiyan, L. (1996). On the Complexity of Dualization of Monotone Disjunctive Normal Forms. *Journal of Algorithms*, 21(3):618–628. [53]

[57] Freeman, L. C. (1977). A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41. [38]

[58] Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239. [38]

[59] Gagneur, J. and Klamt, S. (2004). Computation of elementary modes: a unifying framework and the new binary approach. *BMC bioinformatics*, 5:175. [47]

[60] Gallo, G., Longo, G., Nguyen, S., and Pallottino, S. (1993). Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42:177–201. [129]

[61] Goh, K.-I., Oh, E., Jeong, H., Kahng, B., and Kim, D. (2002). Classification of scale-free networks. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 99(20):12583–12588. [32]

[62] Gomory, R. E. and Hu, T. C. (1961). Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570. [45, 133, 134]

[63] Goyal, S., van der Leij, M. J., Josandeacute, and Moraga-Gonzalez, L. (2006). Economics: An Emerging Small World. *Journal of Political Economy*, 114(2):403–432. [31, 32, 36]

[64] Grafahrend-Belau, E., Schreiber, F., Heiner, M., Sackmann, A., Junker, B. H., Grunwald, S., Speer, A., Winder, K., and Koch, I. (2008). Modularization of biochemical networks based on classification of Petri net t-invariants. *BMC Bioinformatics*, 9(1):90. [17, 18, 129]

[65] Grennan, A. K. (2006). Regulation of starch metabolism in Arabidopsis leaves. *Plant physiology*, 142(4):1343–5. [73]

[66] Hädicke, O. and Klamt, S. (2011). Computing complex metabolic intervention strategies using constrained minimal cut sets. *Metabolic Engineering*, 13:204–213. [52]

[67] Hagen, M. (2008). *Algorithmic and Computational Complexity Issues of MONET*. Cuvillier. [127, 129]

[68] Hao, J. X. and Orlin, J. B. (1992). A faster algorithm for finding the minimum cut in a graph. In *Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, SODA '92, pages 165–174, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics. [45, 135]

[69] Hao, J. X. and Orlin, J. B. (1994). A faster algorithm for finding the minimum cut in a directed graph. *Journal of Algorithms*, 17(3):424–446. [44]

[70] Hardy, S. and Robillard, P. N. (2008). Petri net-based method for the analysis of the dynamics of signal propagation in signaling pathways. *Bioinformatics*, 24(2):209–217. [15, 16]

[71] Hariharan, R., Kavitha, T., Panigrahi, D., and Bhalgat, A. (2007). An $\tilde{O}(mn)$ Gomory-Hu tree construction algorithm for unweighted graphs. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing - STOC '07*, page 605, New York, New York, USA. ACM Press. [133]

[72] Hartman, H. B., Fell, D. A., Rossell, S., Jensen, P. R., Woodward, M. J., Thorndahl, L., Jelsbak, L., Elmerdahl Olsen, J., Raghunathan, A., Daefler, S., and Poolman, M. G. (2014). Identification of potential drug targets in Salmonella enterica sv. Typhimurium using metabolic modelling and experimental validation. *Microbiology*, 160(6):1252–1266. [60]

[73] Haus, U., Klamt, S., and Stephen, T. (2008). Computing Knock-Out Strategies in Metabolic Networks. *J Comput Biol*. [52, 53, 60]

[74] Heiner, M. and Koch, I. (2004). Petri net based model validation in systems biology. In *Applications and Theory of Petri Nets 2004*, pages 216–237. Springer. [129]

[75] Hiller, K. and Metallo, C. M. (2013). Profiling metabolic networks to study cancer metabolism. *Current Opinion in Biotechnology*, 24(1):60–68. [9]

[76] Hochbaum, D. (2008). Graph Algorithms and Network Flows. [142]

[77] Hong-Wu Ma Xue-Ming Zhao, Y.-J. Y. and Zeng, A.-P. (2004). Decomposition of metabolic network into functional modules based on the global connectivity structure of reaction graph. *Bioinformatics*, 20(12):1870–1876. [39]

[78] Hsu, H.-P., Mehra, V., and Grassberger, P. (2003). Structure optimization in an off-lattice protein model. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 68:037703. [130]

[79] Hu, Z., Mellor, J., Wu, J., and DeLisi, C. (2004). VisANT: an online visualization and analysis tool for biological interaction data. *BMC bioinformatics*, 5(1):17. [30]

[80] Jackson, M. O. and Rogers, B. W. (2005). The Economics of Small Worlds. *Journal of the European Economic Association*, 3(2-3):617–627. [28]

[81] Jasmon, G. B. and Foong, K. W. (1987). A Method for Evaluating All the Minimal Cuts of a Graph. *IEEE Transactions on Reliability*, R-36(5):539–545. [46, 141]

[82] Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabási, A. L. (2000). The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654. [13, 18]

[83] Joel, H. B., Douglas, A., and Fred, S. (1997). Hans Krebs and the Puzzle of Cellular Respiration. In *Doing Biology*, chapter 7, pages 72–82. Benjamin Cummings, 1 edition. [19]

[84] Joy, M. P., Brock, A., Ingber, D. E., and Huang, S. (2005). High-betweenness proteins in the yeast protein interaction network. *Journal of biomedicine & biotechnology*, 2005(2):96–103. [38]

[85] Jungreuthmayer, C., Beurton-Aimar, M., and Zanghellini, J. (2013a). Fast computation of minimal cut sets in metabolic networks with a Berge algorithm that utilizes binary bit pattern trees. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 10:1329–33. [53, 71]

[86] Jungreuthmayer, C., Nair, G., Klamt, S., and Zanghellini, J. (2013b). Comparison and improvement of algorithms for computing minimal cut sets. *BMC bioinformatics*, 14(1):318. [52, 53]

[87] Jungreuthmayer, C., Ruckerbauer, D. E., and Zanghellini, J. (2012). Utilizing gene regulatory information to speed up the calculation of elementary flux modes. *ArXiv e-prints*. [54, 145]

[88] Jungreuthmayer, C., Ruckerbauer, D. E., and Zanghellini, J. (2013c). regEfmtool: Speeding up elementary flux mode calculation using transcriptional regulatory rules in the form of three-state logic. *Biosystems*, 113(1):37–39. [68, 69, 145]

[89] Junker, B. H. and Schreiber, F. (2008). *Analysis of Biological Networks*. John Wiley & Sons, Inc., 1 edition. [25, 36, 37]

[90] Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., and Hattori, M. (2004). The KEGG resource for deciphering the genome. *Nucleic Acids Research*, 32(32 Database):D277—-D280. [9]

[91] Karger, D. R. (2000). Minimum cuts in near-linear time. *Journal of Association for Computing Machinery*, 47:46–76. [140]

[92] Karger, D. R. and Levine, M. S. (1998). Finding maximum flows in undirected graphs seems easier than bipartite matching. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC '98*, pages 69–78, New York, New York, USA. ACM Press. [133]

[93] Karger, D. R. and Stein, C. (1996). A new approach to the minimum cut problem. *Journal of the ACM*, 43:601–640. [139]

[94] Karp, G. (2010). Bioenergetics, Enzymes, and Metabolism. In *Cell and Molecular Biology: Concepts and Experiments*, chapter 3, pages 84–112. John Wiley & Sons, Inc., 6 edition. [7]

[95] Karp, P. D., Ouzounis, C. A., Moore-Kochlacs, C., Goldovsky, L., Kaipa, P., Ahrén, D., Tsoka, S., Darzentas, N., Kunin, V., and López-Bigas, N. (2005). Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic acids research*, 33(19):6083–9. [9]

[96] Kauffman, K. J., Prakash, P., and Edwards, J. S. (2003). Advances in flux balance analysis. *Curr Opin Biotechnol*, 14(5):491–496. [12]

[97] Kavvadias, D. J. and Stavropoulos, E. C. (1999). Evaluation of an algorithm for the transversal hypergraph problem. *Algorithm Engineering*, 1668:72–84. [127]

[98] Keseler, I. M., Bonavides-Martínez, C., Collado-Vides, J., Gama-Castro, S., Gunsalus, R. P., Johnson, D. A., Krummenacker, M., Nolan, L. M., Paley, S. M., Paulsen, I. T., Peralta-Gil, M., Santos-Zavaleta, A. D., Shearer, A. G., and Karp, P. D. (2009). EcoCyc: A comprehensive view of Escherichia coli biology. *Nucleic Acids Research*, 37(Database-Issue):464–470. [9]

[99] Klamt, S. (2006). Generalized concept of minimal cut sets in biochemical networks. *Bio Systems*, 83(2-3):233–247. [10, 11, 46, 47, 48, 49, 52, 53, 60]

[100] Klamt, S. and Gilles, E. D. (2004). Minimal cut sets in biochemical reaction networks. *Bioinformatics*, 20(2):226–234. [44, 46, 47, 48, 49, 50, 53, 55, 60]

[101] Klamt, S., Haus, U.-U., and Theis, F. (2009). Hypergraphs and Cellular Networks. *PLoS Comput Biol*, 5(5):e1000385+. [16, 127, 128, 129]

[102] Klamt, S., Saez-Rodriguez, J., and Gilles, E. D. (2007). Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Systems Biology*, 1(1):2. [18, 53, 69]

[103] Klamt, S. and Stelling, J. (2002). Combinatorial complexity of pathway analysis in metabolic networks. *Molecular biology reports*, 29(1-2):233–236. [18, 52]

[104] Koch, I. and Heiner, M. (2008). Petri Nets. In *Analysis of Biological Networks*, chapter 7, pages 139–179. Wiley. [11, 15]

[105] Koch, I., Junker, B. H., and Heiner, M. (2005). Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. *Bioinformatics*, 21(7):1219–1226. [15]

[106] Krajnc, B. and Marhl, M. (2002). The small world in biophysical systems structural properties of glycolysis and the TCA cycle in Escherichia coli. *Cell Mol Biol Lett*, 7(1):129–131. [26]

[107] Krebs, H. A. (1970). The history of the tricarboxylic acid cycle. *Perspectives in Biology and Medicine*, 14(1):154–70. [19, 64]

[108] Larhlimi, A. (2008). *New Concepts and Tools in Constraint-based Analysis of Metabolic Networks*. PhD thesis. [7, 12]

[109] Latora, V. and Marchiori, M. (2002). Is the Boston subway a small-world network? *Physica A: Statistical Mechanics and its Applications*, 314:109–113. [41]

[110] LeBrasseur, N. (2005). The evolution of hubs. *The Journal of Cell Biology*, 170(3):337–337. [77]

[111] Li, G., Höpfner, P., Schäfer, J., Blumenstein, C., Meyer, S., Bostwick, A., Rotenberg, E., Claessen, R., and Hanke, W. (2013). Magnetic order in a frustrated two-dimensional atom lattice at a semiconductor surface. *Nature communications*, 4:1620. [130]

[112] Marie Beurton-Aimar, Nicolas Parisey, François Vallée, Tung V. N. Nguyen, and Sophie Colombié (2011). Metabolite Hubs to Structure Multi-Pathway Networks. (Abstract/Poster). [26]

[113] Marie Beurton-Aimar, Tung V. N. Nguyen, and Sophie Colombié (2012). Analysis of Metabolic Networks Using Minimal Cut Set Computing. (Abstract/Poster). [61]

[114] Martelli, A. (1976). A Gaussian Elimination Algorithm for the Enumeration of Cut Sets in a Graph. *Journal of Association for Computing Machinery*, 23(1):58–73. [141]

[115] Mason, O. and Verwoerd, M. (2007). Graph theory and networks in biology. In *IET Systems Biology*, volume 1, pages 89–119. [25]

[116] Milgram, S. (1967). The Small World Problem. *Psychology Today*, 67(1):61–67. [25, 31]

[117] Montoya, J. M. and Solé, R. V. (2002). Small World Patterns in Food Webs. *Journal of Theoretical Biology*, 214(3):405–412. [32]

[118] Nacher, J. C. and Schwartz, J.-M. (2008). A global view of drug-therapy interactions. *BMC pharmacology*, 8(1):5. [38]

[119] Nagamochi, H. and Ibaraki, T. (1992). A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7(1):583–596. [45, 138]

[120] Nemutlu, E., Zhang, S., Juranic, N. O., Terzic, A., Macura, S., and Dzeja, P. (2012). 18O-assisted dynamic metabolomics for individualized diagnostics and treatment of human diseases. *Croatian medical journal*, 53(6):529–34. [5]

[121] Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, 45:167–256. [31]

[122] Nguyen, T. V. N., Beurton-Aimar, M., and Colombié, S. (2013). Heterotrophic Plant Cell Network Analysis: Comparison Between EFMs and MCSs Methods. (Abstract/Poster). [61]

[123] Nguyen Vu Ngoc, T., Beurton-Aimar, M., and Sophie, C. (2013). Minimal Cut Sets and Its Application to Study Metabolic Pathway Structures. In Patrick Amar, François Képès, and Vic Norris, editors, *Proceedings of the Nice Spring School on Advances in Systems and Synthetic Biology*, pages 71–81, Nice, France. [60]

[124] Nielsen, J. and Villadsen, J. (2002). *Bioreaction engineering principles*. Kluwer Academic/Plenum Publishers, New York, 2^{nd} edition. [8]

[125] Orth, J. D., Thiele, I., and Palsson, B. O. (2010). What is flux balance analysis? *Nat. Biotechnol*, 28(3):245–248. [12, 14, 15]

[126] Ortiz-Pelaez, A., Pfeiffer, D. U., Soares-Magalhães, R. J., and Guitian, F. J. (2006). Use of social network analysis to characterize the pattern of animal movements in the initial phases of the 2001 foot and mouth disease (FMD) epidemic in the UK. *Preventive veterinary medicine*, 76(1-2):40–55. [38]

[127] Palsson, B. (2002). In silico biology through "omics". *Nature biotechnology*, 20:649–650. [14]

[128] Papin, J. A., Price, N. D., and Palsson, B. O. (2002). Extreme Pathway Lengths and Reaction Participation in Genome-Scale Metabolic Networks. *Genome Research*, 12(12):1889–1900. [14]

[129] Papin, J. A., Price, N. D., Wiback, S. J., Fell, D. A., and Palsson, B. O. (2003). Metabolic pathways in the post-genome era. [14]

[130] Papin, J. A., Stelling, J. J., Price, N. D., Klamt, S., Schuster, S., and Palsson, B. O. O. (2004). Comparison of network-based pathway analysis methods. *Trends in Biotechnology*, 22(8):400–405. [12]

[131] Pérès, S. (2005). *Analyse de la structure des réseaux métaboliques: application au métabolisme énergétique mitochondrial*. PhD thesis, Université de Bordeaux 2. [19]

[132] Pérès, S., Beurton-Aimar, M., and Mazat, J. P. (2006). Pathway classification of TCA cycle. *Systems Biology, IEE Proceedings*, 153(5):369–371. [17, 71]

[133] Pérès, S., Beurton-Aimar, M., and Mazat, J. P. M. (2005). Analysis of large set of elementary modes: application to energetic mitochondrial metabolism. In *European Conference on Complex Systems*. [21]

[134] Pérès, S., Felicori, L., and Molina, F. (2013). Elementary flux modes analysis of functional domain networks allows a better metabolic pathway interpretation. *PloS one*, 8(10):e76143. [56]

[135] Pérès, S., Vallée, F., Beurton-Aimar, M., and Mazat, J. P. (2011). ACoM: a classification method for elementary flux modes based on motif finding. *Biosystems*, 103(3):410–419. [18, 56, 70]

[136] Pfeiffer, T., Valdenbro, I. S., Nuño, J. C., Montero, F., and Schuster, S. (1999). METATOOL: for studying metabolic networks. *Bioinformatics*, 15(3):251–257. [18, 66]

[137] Piatek, L. (2011). A new algorithm for generating cuts set in undirected graphs representing electric systems. *PRZEGLAD ELEKTROTECHNICZNY*, 87:123–125. [46]

[138] R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. [35]

[139] Rai, S. (1982). A Cutset Approach to Reliability Evaluation in Communication Networks. *IEEE Transactions on Reliability*, R-31(5):428–431. [43]

[140] Ralf Steuer, Björn H. Junker, Steuer, R., and Junker, B. H. (2008). Computational Models of Metabolism: Stability and Regulation in Metabolic Networks. In *Advances in Chemical Physics*, pages 105–251. John Wiley & Sons, Inc. [12]

[141] Ramachandran, V., Raghuram, A. C., Krishnan, R. V., and Bhaumik, S. K. (2005). *Failure Analysis of Engineering Structures: Methodology and Case Histories*, chapter 5, pages 39–42. ASM International. [43, 142]

[142] Ramadan, E., Tarafdar, A., and Pothen, A. (2004). A hypergraph model for the yeast protein complex network. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, pages 189–. [129]

[143] Rapoport, A. (1951). Nets with distance bias. *The Bulletin of Mathematical Biophysics*, 13(2):85–91. [131]

[144] Rapoport, A. (1953). Spread of information through a population with socio-structural bias: I. Assumption of transitivity. *The Bulletin of Mathematical Biophysics*, 15(4):523–533. []

[145] Rapoport, A. (1957). Contribution to the theory of random and biased nets. *The Bulletin of Mathematical Biophysics*, 19(4):257–277. [131]

[146] Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A. L. (2002). Hierarchical Organization of Modularity in Metabolic Networks. *Science*, 297(5586):1551–1555. [30, 35]

[147] Resat, H., Petzold, L., and Pettigrew, M. F. (2009). Kinetic modeling of biological systems. *Methods in molecular biology (Clifton, N.J.)*, 541:311–35. [12]

[148] Rodríguez, A. and Infante, D. (2009). Network models in the study of metabolism. *Electronic Journal of Biotechnology*, 12(4):1–19. [11]

[149] Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika*, 31(4):581–603. [37, 38]

[150] Sackmann, A., Heiner, M., and Koch, I. (2006). Application of Petri net based analysis techniques to signal transduction pathways. *BMC Bioinformatics*, 7(1):482. [15, 16, 130]

[151] Sangaalofa, T. C. (2012). *Minimal Cut Sets and Their Use in Modelling Flavonoid Metabolism*. PhD thesis, Lincoln University. [46]

[152] Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, 1(1):27–64. [28]

[153] Schilling, C. H., Letscher, D., and Palsson, B. O. (2000). Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *J Theor Biol*, 203(3):229–248. [14]

[154] Schuster, S., Dandekar, T., and Fell, D. A. (1999). Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends in biotechnology*, 17(2):53–60. [47, 65]

[155] 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. *Nature Biotechnology*, 18(3):326–332. [16]

[156] Schuster, S. and Hilgetag, C. (1994). On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems*, 2(2):165–182. [16]

[157] Schuster, S., Hilgetag, C., Woods, J. H., and Fell, D. A. (2002a). Reaction routes in biochemical reaction systems: algebraic properties, validated calculation procedure and example from nucleotide metabolism. *Journal of mathematical biology*, 45(2):153–81. [46]

[158] Schuster, S., Pfeiffer, T., Moldenhauer, F., Koch, I., and Dandekar, T. (2002b). Exploring the pathway structure of metabolism: decomposition into subnetworks and application to Mycoplasma pneumoniae. *Bioinformatics*, 18(2):351–61. [46]

[159] Sen, P., Dasgupta, S., Chatterjee, A., Sreeram, P. A., Mukherjee, G., and Manna, S. S. (2003). Small-world properties of the Indian railway network. *Phys. Rev. E*, 67(3):36106. [31]

[160] Sharafat, A. R. and Márouzi, O. R. (2002). A novel and efficient algorithm for scanning all minimal cutsets of a graph. [140]

[161] Shen, Y. (1995). A new simple algorithm for enumerating all minimal paths and cuts of a graph. *Microelectronics Reliability*, 35(6):973–976. [46]

[162] Shi, J. and Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 22(8):888–905. [143]

[163] Steuer, R., Gross, T., Selbig, J., and Blasius, B. (2006). Structural kinetic modeling of metabolic networks. *Proceedings of the National Academy of Sciences*, 103(32):11868–11873. [12]

[164] Stoer, M. and Wagner, F. (1997). A Simple Min Cut Algorithm. *Journal of the ACM*, 44(4):585–591. [45, 46, 138]

[165] Supekar, K., Menon, V., Rubin, D., Musen, M., and Greicius, M. D. (2008). Network analysis of intrinsic functional brain connectivity in Alzheimer's disease. *PLoS computational biology*, 4(6):e1000100. [25]

[166] Tan, Z. (2003). Minimal cut sets of s-t networks with k-out-of-n nodes. *Reliability Engineering and System Safety*, 82:49–54. [45]

[167] Terzer, M. and Stelling, J. (2008). Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics*, 24(19):2229–2235. [18, 54]

[168] Trinh, C., Wlaschin, A., and Srienc, F. (2009). Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism. *Applied Microbiology and Biotechnology*, 81(5):813–826. [47]

[169] Trinh, C. T., Carlson, R., Wlaschin, A., and Srienc, F. (2006). Design, construction and performance of the most efficient biomass producing E. coli bacterium. *Metabolic engineering*, 8(6):628–38. [52]

[170] Trinh, C. T. and Thompson, R. A. (2012). Elementary mode analysis: a useful metabolic pathway analysis tool for reprograming microbial metabolic pathways. *Sub-cellular biochemistry*, 64:21–42. [12]

[171] Tsukiyama, S., Shirakawa, I., Ozaki, H., and Ariyoshi, H. (1980). An algorithm to enumerate all cutsets of a graph in linear time per cutset. *Journal of Association for Computing Machinery*, 27(4):619–632. [140, 141]

[172] Unrean, P., Trinh, C. T., and Srienc, F. (2010). Rational design and construction of an efficient E. coli for production of diapolycopendioic acid. *Metabolic engineering*, 12(2):112–22. [52]

[173] Varma, A. and Palsson, B. O. (1994). Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. [9]

[174] Vendruscolo, M., Dokholyan, N. V., Paci, E., and Karplus, M. (2002). Small-world view of the amino acids that play a key role in protein folding. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 65(6 Pt 1):061910. [35]

[175] Verkhedkar, K. D., Raman, K., Chandra, N. R., and Vishveshwara, S. (2007). Metabolome based reaction graphs of M. tuberculosis and M. leprae: a comparative network analysis. *PloS one*, 2(9):e881. [77]

[176] von Kamp, A. and Klamt, S. (2014). Enumeration of smallest intervention strategies in genome-scale metabolic networks. *PLoS computational biology*, 10(1):e1003378. [46, 47, 80]

[177] Wagner, A. and Fell, D. A. (2001). The small world inside large metabolic networks. In *Proceedings of the Conference of The Royal Society in London B*, volume 268, pages 1803–1810. [13, 26, 35]

[178] Watts, D. J. (1999). *Small worlds: The dynamics of networks between order and randomness*. Princeton University Press, Princeton, NJ. [32]

[179] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393:440–442. [25, 28, 31, 35]

[180] Whitaker, J. W., Letunic, I., McConkey, G. A., and Westhead, D. R. (2009). metaTIGER: a metabolic evolution resource. *Nucleic Acids Research*, 37(Database-Issue):531–538. [9]

[181] Whitney, H. (1933). Planar Graphs. *Fund. Math.*, 21:73–84. [43]

[182] Wilhelm, T., Behre, J., and Schuster, S. (2004). Analysis of structural robustness of metabolic networks. *Systems biology*, 1(1):114–120. [48]

[183] Wright, B. E., Butler, M. H., and Albe, K. R. (1992). Systems analysis of the tricarboxylic acid cycle in Dictyostelium discoideum. I. The basis for model construction. *The Journal of biological chemistry*, 267(5):3101–5. [19]

[184] Yuang, M. C., Chen, Y. G., and Yen, M. T. (1995). Optimal multicast routing for ATM networks. In *Local Computer Networks, 1995., Proceedings. 20th Conference on*, pages 413–421. [43]

[185] Zevedei-Oancea, I. and Schuster, S. (2003). Topological analysis of metabolic networks based on Petri net theory. *In silico biology*, 3(3):29. [16]

[186] Zhu, X., Gerstein, M., and Snyder, M. (2007). Getting connected: analysis and principles of biological networks. *Genes Dev*, 21(9):1010–1024. [30]

[187] Zio, E. (2007). *An introduction to the basics of reliability and risk analysis*, volume 13 of *Quality, Reliability and Engineering Statistics*, chapter 7, pages 128–132. World Scientific Publishing Co. Pte. Ltd. [43, 142]

[188] Zubay, G. (1993). *Biochemistry*. Wm. C. Brown, Oxford, 3ˆ{rd} edition. [8]

Index

Acronyms

cMCSs Constraint MCSs. 52

CNA CellNetAnalyzer. 66, 69

EFM Elementary Flux Mode. 16

EFMs Elementary Flux Modes. 14, 16, 54, 58, 60, 87, 89, 90

FBA Flux Balance Analysis. 14, 16, 64

KEGG Kyoto Encyclopedia of Genes and Genomes. 8

MCS Minimal Cut Set. 46

MCSs Minimal Cut Sets. 55, 58, 60, 87, 89, 90

MMF Minimal Metabolic Functionality. 52

MNHPC Metabolic Network of Heterotrophic Plant Cells. 21, 35, 59, 60, 64, 67, 68, 85

MPA Metabolic Pathway Analysis. 46, 60

ODEs ordinary differential equations. 12

SCCs strong connected components. 32, 36

SFNs scale-free networks. 31, 36

SWN small-world networks. 36

SWN small-world network. 31, 32, 35

SWNs Small-world networks. 30, 31

TCA TriCarboxylic Acid. 58

List of Abbreviations

Abbreviation	Meaning
Ac	Acetate
AccoA	acetyl coenzymeA
ADPG	ADP-glucose
aKG	2-oxoglutarate
Ala	alanine
Asp	aspartate
CellWall	cell wall polysaccharides
Cit	citrate
DAG	Diacyl glycerol
DHA	Dihydroxyacetone
DHAP	Dihydroxyacetone phosphate
dPG	diphospho glycerate
3PG	3-phospho glycerate
2PG	2-phospho glycerate
E4P	erythrose 4-phosphate
FA	fatty acids
FA16 and FA18	fatty acids with a carbon chain of 16 and 18 carbon respectively
FADH2	flavin adenine dinucleotide
F6P	fructose-6-phosphate
Fru	fructose
F16bP	fructose-1,6-biphosphate
Fum	fumarate
GAP	glyceraldehyde-3-phosphate
Glc	glucose
Gln	glutamine
Glu	glutamate
Glyc	glycerol
Glyc3P	glycerol 3-phosphate

Abbreviation	Meaning
G1P	glucose-1-phosphate
G6P	glucose-6-phosphate
Icit	isocitrate
Lac	Lactate
Lys	Lysine
Mal	malate
NADH	nicotinamide adenine dinucleotide
NADPH	nicotinamide adenine dinucleotide phosphate
NMR	nuclear magnetic resonance
OAA	oxaloacetate
PEP	phosphoenolpyruvate
Pyr	pyruvate
P5P	Pentoses-5-phosphate
Ri5P	ribose-5-phosphate
Ru5P	ribulose-5-phosphate
suc	sucrose
succ	succinate
sucP	sucrose phosphate
S7P	sedoheptulose-7-phosphate
Trehal	Trehalose
UDP_Glc	Uridine diphosphate-glucose
UTP	Uridine triphosphate
X5P	xylulose-5-phosphate
'_in'	for exogenous metabolites uptake by the cell (glucose, glutamine and alanine)
'_out'	for exogenous amino acids (glutamine, glutamate, aspartate, alanine)
'_p'	for metabolites located in plastid
'_v'	for metabolites located invacuole

List of Figures

List of Tables

Appendix

Appendix A

Data Descriptions

The data presented here are formed in METATOOL format.

A.1 TCA cycle

```
-ENZREV
R9 R12 R13 R14 R15 T1 T2 T5 T7

-ENZIRREV
R6i R7i R8i R10i R11i T6

-METINT
OAA ACoA Cit Akg SucCoA Succ Fum Mal Isocit  Pi Pyr

-METEXT
Pyr_ext NAD NADH2 FAD FADH2  CoA ADP ATP  H2O CO2 Mal_ext Cit_ext AKG_ext Pi_ext

-CAT

R6i : Pyr + CO2 + ATP = OAA + Pi + ADP .
R7i : Pyr + NAD + CoA = ACoA + NADH2 + CO2 .
R8i : OAA + ACoA + H2O = Cit + CoA .
R9 : Cit = Isocit .
R10i : Isocit + NAD = Akg + NADH2 + CO2 .
R11i : Akg + NAD + CoA = SucCoA + NADH2 + CO2 .
R12 : SucCoA + Pi + ADP = Succ + CoA + ATP .
R13 : Succ + FAD = Fum + FADH2 .
R14 : Fum + H2O = Mal .
R15 : Mal + NAD = OAA + NADH2 .
T1 : Cit + Mal_ext = Mal + Cit_ext .
T2 : AKG_ext + Mal = Mal_ext + Akg .
T5 : Pi_ext = Pi .
T6 : Pyr_ext = Pyr .
T7 : Mal + Pi_ext = Pi + Mal_ext .
```

A.2 Muscle

```
-ENZREV
R3r R4 R5 R9 R12 R14 R15 R16 R17 R24 R27  R30 R32 T1r T2r T3 T4 T5 T7  T9 T10
T11 T12 T13 T19

-ENZIRREV
R1i R2i R6i R7i R8i R10i R11i R13i R28i R31i T6i T20i

-METINT
OAA AcCoA Cit AKG SucCoA Succ Fum Mal Isocit  AcetoAcCoA Acetoacetate  AcylCoA
ATP ADP AMP Pi_M Pi2_M Pyr HB Acylcarnitine NADH CoA NAD FAD FADH2 FADm FADH2m
Carnitine Asp Glu H

-METEXT
Glu_ext CO2_ext AKG_ext Fum_ext Mal_ext  ADP_ext ATP_ext  Pi_M_ext Pi2_M_ext
Pyr_ext Cit_ext HB_ext  AA_ext Carnitine_ext  NH3_ext Asp_ext H_ext
AcylCarnitine_ext Glycerol-3-P_ext Dihydroxy-acetone-P_ext H2O_ext

-CAT
R1i : NADH  + 10 H = NAD + 10 H_ext .
R2i : FADH2m + 6 H = FADm + 6 H_ext .
R3r : ADP + Pi_M + 3 H_ext = ATP + 3 H .
R4 : ATP + AMP = 2 ADP .
R5 : Pi2_M + H = Pi_M .
R6i : Pyr + CO2_ext + ATP = OAA + Pi_M + ADP .
R7i : Pyr + NAD + CoA = AcCoA + NADH + H + CO2_ext .
R8i : OAA + AcCoA = Cit + CoA .
R9 : Cit = Isocit .
R10i : Isocit + NAD = AKG + NADH + H + CO2_ext .
R11i : AKG + NAD + CoA = SucCoA + NADH + H + CO2_ext .
R12 : SucCoA + Pi_M + ADP = Succ + CoA + ATP .
R13i : Succ + FADm = Fum + FADH2m .
R14 : Fum = Mal .
R15 : Mal + NAD = OAA + NADH + H .
R16 : Glu + NAD + H2O_ext =  AKG + NH3_ext + NADH + H .
R17 : OAA + Glu = Asp + AKG .
R27 : Acetoacetate + NADH + H = HB + NAD .
R28i : SucCoA + Acetoacetate = AcetoAcCoA + Succ .
R24 : CoA + AcetoAcCoA = 2 AcCoA .
R30 :  Acylcarnitine + CoA = AcylCoA + Carnitine .
R31i : AcylCoA + 7 FAD + 7 NAD + 7 CoA = 7 NADH + H + 7 FADH2 + 8 AcCoA .
T1r : Cit + H + Mal_ext = Mal + Cit_ext + H_ext .
T2r : Mal_ext + AKG = AKG_ext + Mal .
T3 : AcylCarnitine_ext + Carnitine = Carnitine_ext + Acylcarnitine .
T4 : ADP_ext + ATP = ADP + ATP_ext .
```

```
T5 : Pi_M_ext + H_ext = Pi_M + H .
T6i : Pyr_ext + H_ext = Pyr + H .
T7 : Pi2_M + Mal_ext = Mal + Pi2_M_ext .
T9 : H_ext = H .
T10 : HB = HB_ext .
T11 : AA_ext = Acetoacetate .
T12 : Asp + Glu_ext + H_ext = Asp_ext + Glu + H .
T13 : Fum_ext + Mal = Fum + Mal_ext .
T19 : AKG + Pi2_M_ext = AKG_ext + Pi2_M .
T20i : Glu_ext + H_ext = Glu + H .
R32 : Glycerol-3-P_ext + FAD = Dihydroxy-acetone-P_ext + FADH2 .
```

A.3 Liver

```
-ENZREV
R3r R4 R5 R9 R12 R14 R15 R16 R17 R24 R27 R30  R32 T1r T2r T3 T4 T5 T7 T8 T9 T10
T11 T12 T13 T19 T21 T22

-ENZIRREV
R1i R2i R6i R7i R8i R10i R11i R13i  R21i R22i R23i R25i R26i R31i T6i T20i

-METINT
OAA AcCoA Cit AKG SucCoA Succ Fum Mal Isocit  CarbamoylP AcetoAcCoA HMGCoA
Acetoacetate  AcylCoA  ATP ADP AMP Pi- Pi2- Pyr Ornit Citrulline HB
Acylcarnitine NADH CoA NAD FAD FADH2 FADm FADH2m Carnitine Asp Glu H Glutamine

-METEXT
Glu_ext CO2 AKG_ext Fum_ext Mal_ext  ADP_ext ATP_ext  Pi-_ext Pi2-_ext Pyr_ext
Cit_ext Citru_ext Ornit_ext HB_ext   AA_ext Carnitine_ext  NH3 Asp_ext H_ext
Glutamine_ext AcylCarnitine_ext Glycerol-3-P_ext Dihydroxy-acetone-P_ext
Ornit_ext H2O_ext

-CAT
R1i : NADH  + 10 H = NAD + 10 H_ext .
R2i : FADH2m + 6 H = FADm + 6 H_ext .
R3r : ADP + Pi- + 3 H_ext = ATP + 3 H .
R4 : ATP + AMP = 2 ADP .
R5 : Pi2- + H = Pi- .

R6i : Pyr + CO2 + ATP = OAA + Pi- + ADP .
R7i : Pyr + NAD + CoA = AcCoA + NADH + H + CO2 .
R8i : OAA + AcCoA = Cit + CoA .
R9 : Cit = Isocit .
R10i : Isocit + NAD = AKG + NADH + H + CO2 .
R11i : AKG + NAD + CoA = SucCoA + NADH + H + CO2 .
R12 : SucCoA + Pi- + ADP = Succ + CoA + ATP .
```

```
R13i : Succ + FADm = Fum + FADH2m .
R14 : Fum = Mal .
R15 : Mal + NAD = OAA + NADH + H .
R16 : Glu + NAD + H2O_ext =  AKG + NH3 + NADH + H .
R17 : OAA + Glu = Asp + AKG .

R21i : 2 ATP + NH3 + CO2 + H2O_ext = 2 ADP + Pi- + CarbamoylP .
R22i : CarbamoylP + Ornit = Pi- + Citrulline .
R23i : Glutamine + H2O_ext = Glu + NH3 .

R24 : 2 AcCoA = CoA + AcetoAcCoA .
R25i : AcCoA + AcetoAcCoA = HMGCoA + CoA .
R26i : HMGCoA = AcCoA + Acetoacetate .
R27 : Acetoacetate + NADH + H = HB + NAD .

R30 :  Acylcarnitine + CoA = AcylCoA + Carnitine .
R31i : AcylCoA + 7 FAD + 7 NAD + 7 CoA = 7 NADH + H + 7 FADH2 + 8 AcCoA .

T1r : Cit + H + Mal_ext = Mal + Cit_ext + H_ext .
T2r : Mal_ext + AKG = AKG_ext + Mal .
T3 : AcylCarnitine_ext + Carnitine = Carnitine_ext + Acylcarnitine .
T4 : ADP_ext + ATP = ADP + ATP_ext .
T5 : Pi-_ext + H_ext = Pi- + H .
T6i : Pyr_ext + H_ext = Pyr + H .
T7 : Pi2- + Mal_ext = Mal + Pi2-_ext .
T8 : Citrulline + H + Ornit_ext = Citru_ext + H_ext + Ornit .
T9 : H_ext = H .
T10 : HB = HB_ext .
T11 : AA_ext = Acetoacetate .
T12 : Asp + Glu_ext + H_ext = Asp_ext + Glu + H .
T13 : Fum_ext + Mal = Fum + Mal_ext .
T19 : AKG + Pi2-_ext = AKG_ext + Pi2- .
T20i : Glu_ext + H_ext = Glu + H .
T21 : Ornit_ext + H = Ornit + H_ext .
T22 : Glutamine_ext = Glutamine  .
R32 : Glycerol-3-P_ext + FAD = Dihydroxy-acetone-P_ext + FADH2  .
```

A.4 MNHPC

```
-ENZREV
Vpgi Vald Vtpi Vmdh Vpgi_p Vepi_p Tg6p Ttp Tpep Vsusy Vala Vgdh
Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh Vriso_p_Vtkx_p_Vtald_p

-ENZIRREV
Glc_up ala_up gln_up_Vgs Vg6pdh_Vepi_Tx5p Vpfk_p_Vald_p_Vtpi_p
```

```
Vpk_p_Vpdh_p_VFAx_Vdag_Vglyc3P Vat_Vss_Vpglm_p Vut_NRJ3_Vpglm Vsps_Vspace
NRJ2_Vkgdh_Vsdh_Vfum Vasp_out_Vasp Vhk1 Vhk2 Vpfk Vfbp Vpk Vcl Vpepc Vpdh Vcs
Vg6pdh_p Vgapdh_p Vrbco Vme Vinv Vcw NRJ1 NRJ1b Vgl_out Vala_out Vac_g Vac_f
Vac_s Vac_m Vac_c

-METINT
Suc Glc G6P F6P F16bP DHAP GAP PEP OAA AccoA cit aKG mal pyr Ru5P F6P_p G6P_p
DHAP_p Ru5P_p X5P_p PEP_p UDPG Fru ATP NADH NADPH  ala glu

-METEXT
Glc_in gln_in ala_in CellWall DAG starch ala_out asp_out gl_out CO2 Glc_v Fru_v
Suc_v mal_v cit_v

-CAT
Glc_up : Glc_in = Glc .
ala_up : ala_in = ala .
gln_up_Vgs : aKG + NADPH + gln_in = 2 glu .
Vhk1 : Fru + ATP = F6P .
Vhk2 : Glc + ATP = G6P .
Vpfk : F6P + ATP = F16bP .
Vfbp : F16bP = F6P .
Vpk : PEP = pyr + ATP .
Vcl : cit + ATP = OAA + AccoA .
Vpepc : PEP + CO2 = OAA .
Vpdh : pyr = AccoA + NADH + CO2 .
Vcs : OAA + AccoA = cit .
Vg6pdh_Vepi_Tx5p : G6P = X5P_p + NADPH + CO2 .
Vg6pdh_p : G6P_p = Ru5P_p + NADPH + CO2 .
Vpfk_p_Vald_p_Vtpi_p : F6P_p + ATP = 2 DHAP_p .
Vgapdh_p : DHAP_p = PEP_p + ATP + NADH .
Vpk_p_Vpdh_p_VFAx_Vdag_Vglyc3P : 4 AccoA + 3 DHAP_p + 48 PEP_p + 4 ATP + 88
NADPH = 45 NADH + 3 DAG + 48 CO2 .
Vrbco : Ru5P_p + CO2 = 2 DHAP_p .
Vme : mal = pyr + NADH + CO2 .
Vat_Vss_Vpglm_p : G6P_p + ATP = starch .
Vut_NRJ3_Vpglm : G6P + ATP = UDPG .
Vinv : Suc = Glc + Fru .
Vsps_Vspace : F6P + UDPG = Suc .
Vcw : UDPG = CellWall .
NRJ1 : NADH = 2 ATP .
NRJ1b : 2 NADPH = 3 ATP .
NRJ2_Vkgdh_Vsdh_Vfum : aKG = mal + 2 ATP + NADH + CO2 .
Vgl_out : glu = gl_out .
Vasp_out_Vasp : OAA + glu = aKG + asp_out .
Vala_out : ala = ala_out .
Vac_g : Glc = Glc_v .
```

```
Vac_f : Fru = Fru_v .
Vac_s : Suc = Suc_v .
Vac_m : mal = mal_v .
Vac_c : cit = cit_v .
Vpgi : G6P = F6P .
Vald : F16bP = DHAP + GAP .
Vtpi : DHAP = GAP .
Vgapdh_Vpgk_Vpgm_Veno : GAP = PEP + ATP + NADH .
Vaco_Vidh : cit = aKG + NADH + CO2 .
Vmdh : mal = OAA + NADH .
Vpgi_p : G6P_p = F6P_p .
Vepi_p : Ru5P_p = X5P_p .
Vriso_p_Vtkx_p_Vtald_p : Ru5P_p + 2 X5P_p = 2 F6P_p + DHAP_p .
Tg6p : G6P = G6P_p .
Ttp : DHAP = DHAP_p .
Tpep : PEP = PEP_p .
Vsusy : UDPG + Fru = Suc .
Vala : pyr + glu = aKG + ala .
Vgdh : aKG + NADH = glu .
```

A.5 Aracell

```
-ENZREV
Vsusy Vpgi Vald Vtpi Vgapdh Vpgm Veno Vaco Vidh Vkgdh Vsdh Vfum Vmdh Vald_p
Vpgi_p Vtk1_p Vtk2_p Vtald_p Tg6p Ttp Tg1p Tpep Tp5p Vpglm Vpglm_p Vala Vgdh

-ENZIRREV
Suc_up ala_up gln_up Vinv Vsps Vspase Vhk Vgk Vpfk Vfbp Vpgk Vpk Vcl Vpepc Vpdh
Vcs Vicl Vms Vg6pdh Vg6pdh_p Vpfk_p Vgapdh_p Vpk_p Vpdh_p Vrbco Vac_s Vac_m Vme
Vpepck Vat Vss Vsd Vut Vgs Vasp Vser Vgly Vgdc Vhis Vshik Vtyr Vphe Vval Vile
Vhser Vthr Vleu Vlys Vasn VCO2 Vtag VCW Vala_out Vasp_out Vglu_out Vgln_out
Vser_out VFA16 VFA18 VFA NRJ1 NRJ1b NRJ2 NRJ3 Tpyr

-METINT
Suc Glc G6P F6P F16bP DHAP GAP dPG 3PG 2PG PEP OAA AccoA cit icit aKG succ fum
mal Gox pyr P5P F6P_p G6P_p F16bP_p TP_p P5P_p S7P_p E4P_p PEP_p pyr_p AccoA_p
UDPG ADPG G1P G1P_p SucP Fru ala asp FA16 FA18 tag glu gln shik ser gly hser

-METEXT
Suc_in gln_in ala_in Suc_v mal_v CO2 CellWall ala_out asp_out glu_out gln_out
ser_out FA starch C1 his tyr phe val ile thr leu lys asn ATP UTP NADH NADPH
FADH2

-CAT
Suc_up : Suc_in = Suc .
ala_up : ala_in = ala .
```

```
gln_up : gln_in = gln .
Vhk : ATP + Fru = F6P .
Vgk : ATP + Glc = G6P .
Vinv : Suc = Fru + Glc .
Vsusy : Fru + UDPG = Suc .
Vsps : F6P + UDPG = SucP .
Vspase : SucP = Suc .
Vpgi : G6P = F6P .
Vpfk : ATP + F6P = F16bP .
Vfbp : F16bP = F6P .
Vald : F16bP = DHAP + GAP .
Vtpi : DHAP = GAP .
Vgapdh : GAP = dPG + NADH .
Vpgk : dPG = 3PG + ATP .
Vpgm : 3PG = 2PG .
Veno : 2PG = PEP .
Vpk : PEP = ATP + pyr .
Vpepc : CO2 + PEP = OAA .
Vpdh : pyr = CO2 + AccoA + NADH .
Vcl : cit + ATP = OAA + AccoA .
Vcs : OAA + AccoA = cit .
Vaco : cit = icit .
Vidh : icit = aKG + CO2 + NADH .
Vkgdh : aKG = CO2 + NADH + succ .
Vsdh : succ = FADH2 + fum .
Vfum : fum = mal .
Vmdh : mal = NADH + OAA .
Vicl : icit = succ + Gox .
Vms : AccoA  + Gox = mal .
Vme : mal = CO2 + NADH + pyr .
Vpepck : OAA + ATP = CO2 + PEP .
Vg6pdh : G6P = CO2 + P5P + NADPH .
Vpgi_p : G6P_p = F6P_p .
Vg6pdh_p : G6P_p = CO2 + P5P_p + NADPH .
Vtk1_p : S7P_p + TP_p = 2 P5P_p .
Vtk2_p : F6P_p + TP_p = E4P_p + P5P_p .
Vtald_p : S7P_p + TP_p = E4P_p + F6P_p .
Vpfk_p : ATP + F6P_p = F16bP_p .
Vald_p : F16bP_p = 2 TP_p .
Vgapdh_p : TP_p = ATP + NADH + PEP_p .
Vpk_p : PEP_p = ATP + pyr_p .
Vpdh_p : pyr_p = CO2 + AccoA_p + NADH .
Vrbco : P5P_p + CO2 = 2 TP_p .
Vac_s : Suc = Suc_v .
Vac_m : mal = mal_v .
Tg6p : G6P = G6P_p .
```

```
Ttp : GAP = TP_p .
Tg1p : G1P = G1P_p .
Tpep : PEP = PEP_p .
Tpyr : pyr = pyr_p .
Tp5p : P5P = P5P_p .
Vpglm_p : G1P_p = G6P_p .
Vat : G1P_p + ATP = ADPG .
Vss : ADPG = starch .
Vsd : starch = G1P_p .
Vpglm : G1P = G6P .
Vut : G1P + UTP = UDPG .
Vala : pyr + glu = ala + aKG .
Vasp : OAA + glu = asp + aKG .
Vgdh : aKG + NADH = glu .
Vgs : gln + aKG + NADPH = 2 glu .
Vser : TP_p + glu = aKG + NADH + ser .
Vgly : ser = gly + C1 .
Vgdc : gly = CO2 + C1 .
Vhis : gln + P5P_p + ATP = aKG + 2 NADH + his .
Vshik : E4P_p + PEP_p + ATP + NADPH = shik .
Vtyr : shik + glu = aKG + NADH + CO2 + tyr .
Vphe : shik + glu = aKG + CO2 + phe .
Vval : 2 pyr_p + NADPH + glu =  aKG + CO2 + val .
Vile : pyr_p + NADPH + glu + thr =  aKG + CO2 + ile .
Vhser : asp + NADPH = hser .
Vthr : hser + ATP = thr .
Vleu : 2 pyr_p + AccoA + NADPH + glu = aKG + NADH + CO2 + leu .
Vlys : OAA + pyr_p + NADPH = CO2 + lys .
Vasn : 2 ATP + asp = asn .
Vtag : TP_p + ATP = tag .
Vala_out : ala  = ala_out .
Vasp_out : asp = asp_out .
Vglu_out : glu = glu_out .
Vgln_out : gln = gln_out .
Vser_out : ser = ser_out .
VCW : UDPG = CellWall .
VFA16 : 8 AccoA_p + 7 ATP + 14 NADPH = FA16 .
VFA18 : AccoA + FA16 + ATP + NADPH = FA18 .
VFA : FA16 + 2 FA18 = 3 FA .
NRJ1 :  NADH = 2 ATP .
NRJ1b : 2 NADPH = 3 ATP .
NRJ2 : FADH2 = 2 ATP .
NRJ3 : ATP = UTP .
```

Appendix B

Implementation

Petri net concepts provide additional tools for the modelling of metabolic networks. Here the similarities between the counterparts in traditional biochemical modelling and Petri net theory are discussed. For example the stoichiometric matrix of a metabolic network corresponds to the incidence matrix of the Petri net. The flux modes and conservation relations have the T-invariants, respectively, P-invariants as counterparts. We reveal the biological meaning of some notions specific to the Petri net framework (traps, siphons, deadlocks, liveness). We focus on the topological analysis rather than on the analysis of the dynamic behaviour. The treatment of external metabolites is discussed. Some simple theoretical examples are presented for illustration. Also the Petri nets corresponding to some biochemical networks are built to support our results. For example, the role of triose phosphate isomerase (TPI) in Trypanosoma brucei metabolism is evaluated by detecting siphons and traps. All Petri net properties treated in this contribution are exemplified on a system extracted from nucleotide metabolism.

B.1 Organism studied: *Brassica napus*

To focus on the structural functions of specific reactions and the interactions between the components of the metabolic network and how these interactions give rise to the function and behaviour, an appropriate coherent self-contained sub network must be extracted from the genome scale metabolic network of an appropriate organism. The standard plant system, *Brassica napus* (also shortly called *B. napus*) embryos, is used in this research because of the reasons set out below.

B. napus is an annual or biennial herbaceous plant in the *Brassicaceae* that belonging to the cabbage or mustard family.

Information for the research is obtained from the AraCyc database, available online and containing the full metabolic network of the *B. napus* embryos.

B.2 Our general protocol

Systematically, we represent our approach in the step-by-step instructions as shown in Figure B.2.1.

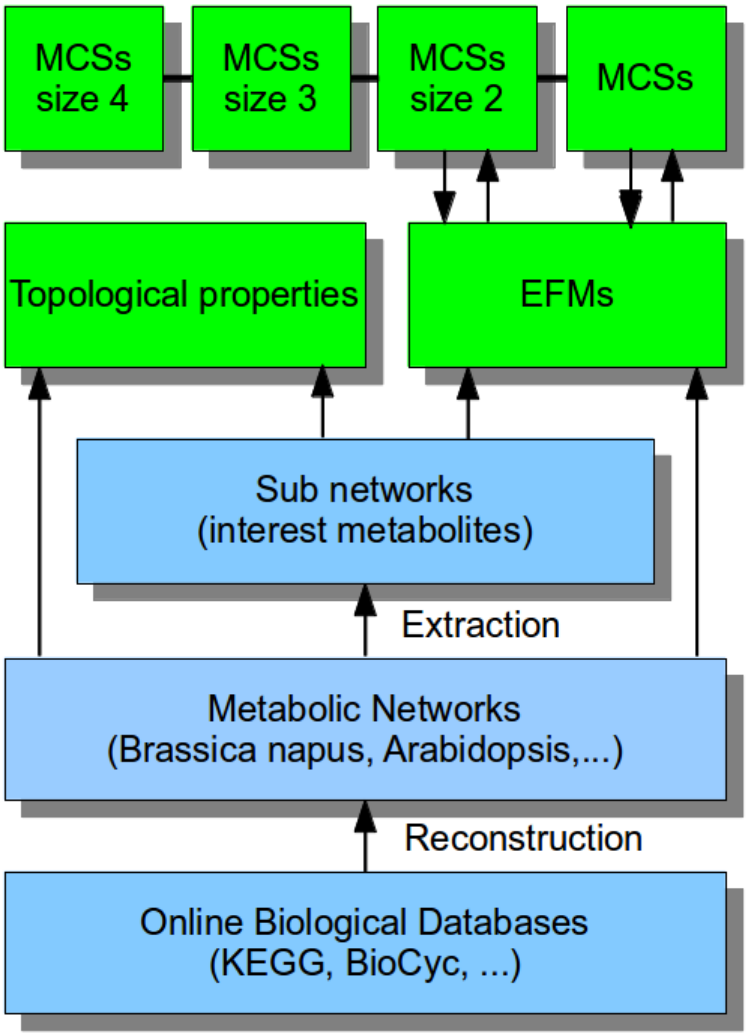


Figure B.2.1: Computational model

B.3 Explanation of the model

Our model divided into 5 steps can be explained as follows:

Reconstruction starts rebuilding the metabolic network of a studied organism from one of online biological databases.

Extraction aims to get a part of the whole network. Sub networks which extracted from the complete one enable to see deeply the studied organism beside the global view.

Computation of topological properties is the step of computing coherent structural measures. The aim of this step is to study the networks' size.

In Chapter 4, we shall show the results obtained in the specific application on a plant cell metabolism.

B.4 Strategies of computing EFMs and MCSs

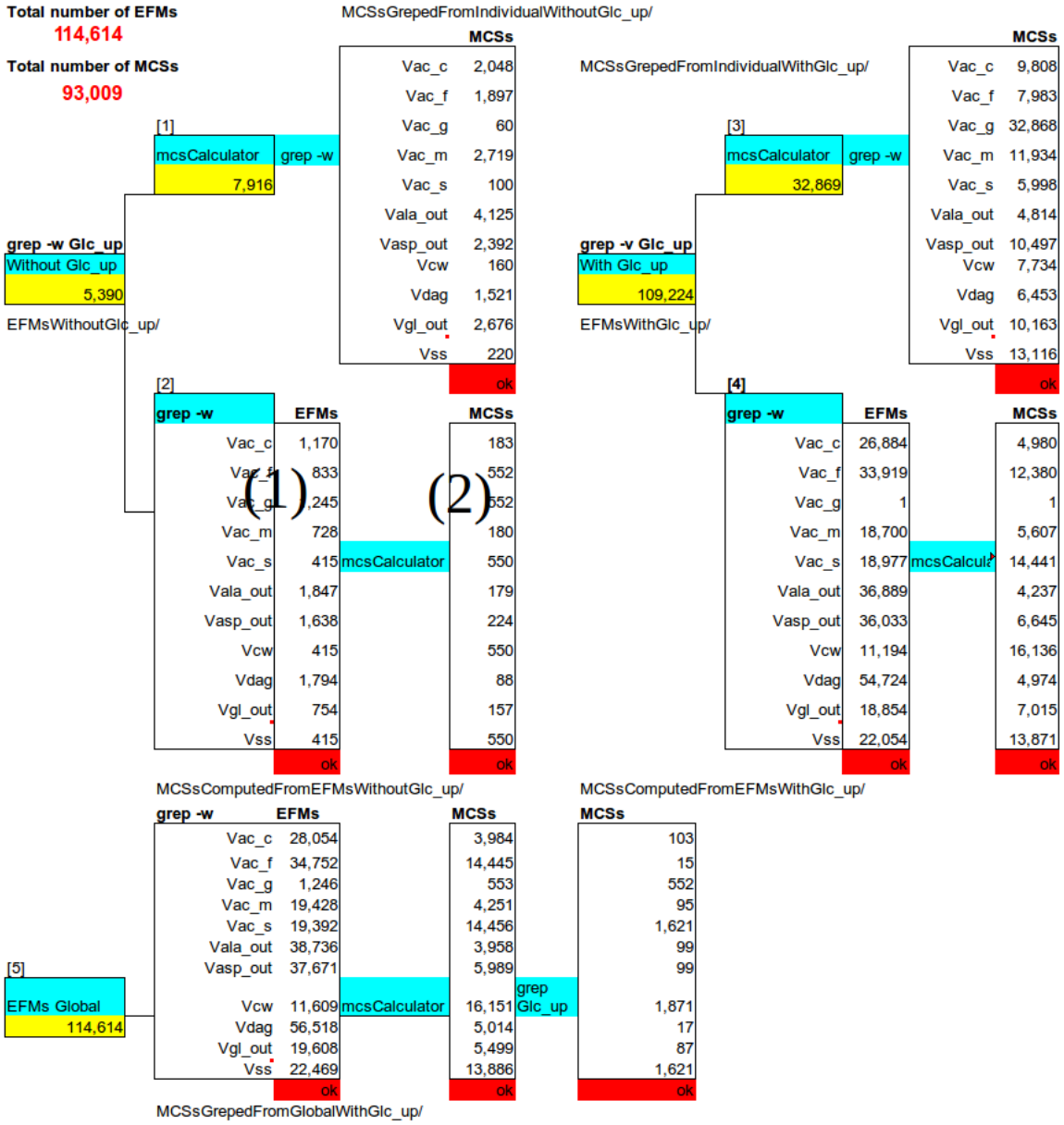


Figure B.4.2: Strategies of computing EFMs and MCSs

B.5 Computing tools

In order to do all experiments in this research, we have used some computing tools as well as programming resources as follows:

- Metatool
- CellNetAnalyzer

- efmtool
- regEfmtool
- Matlab
- R programming language
- C++ programming language
- igraph package for R
- networkx package for Python
- lemon package for C++

Appendix C

Methods and models from Graph Theory

C.1 Hypergraphs

hypergraphs. Indeed, Bollobás introduced hypergraphs in 1986 [24]. For more theoretical issues of hypergraph, we can refer to Berge's works [20], who was also known as the person of the first people giving the attempt to solve the minimal transversal problem of hypergraphs. Though there are some contributions of researching community on this problem such as [41, 97] with the purpose of finding the best methods in computation of minimal transversal sets. *hyperedge*

Definition C.1 (Hypergraph [20])**.** *A hypergraph $\mathcal{H}$ can be defined as a pair (V, E), where V is a set of vertices, and E is a set of hyperedges between the vertices. Each hyperedge is a set of vertices: $E \in \{\{u, v, ...\} \in 2V\}$.*

Example C.1. An example[1] of a hypergraph as Figure C.1.1a, with X $= \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ and E $= \{e_1, e_2, e_3, e_\} = \{\{v_1, v_2, v_3\}, \{v_2, v_3\}, \{v_3, v_5, v_6\}, \{v_4\}\}$.

Example C.2. The another example is extracted from Klamt et al. [101] as Figure C.1.1b. This hypergraph has 5 vertices and 3 edges. From the original system, we can represent it under a hypergraph or a classical graph.

Let us finish this section on hypergraphs by defining some further notations. A very important notion is that of transversals - also often called hitting sets.

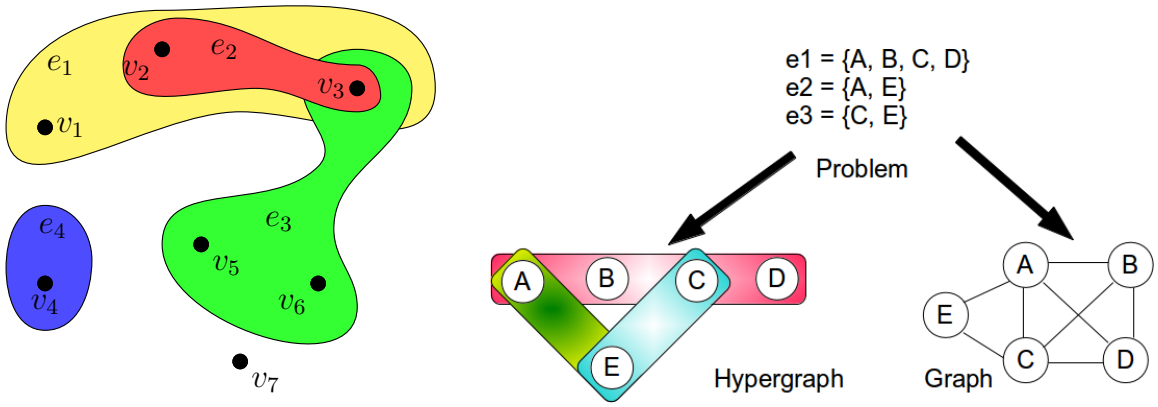
Definition C.2 ((Minimal) Transversal)**.** *A transversal of a hypergraph $\mathcal{H}$ is a vertex set $t \subseteq V$ that has a non-empty intersection with each edge of $\mathcal{H}$. A transversal t is minimal if and only if no proper subset of t is a transversal.*

Thus, just as hypergraphs are a generalisation of graphs, transversals generalise the notion of vertex covers. Note that the set of all minimal transversals also is a hypergraph.

Definition C.3 (Transversal Hypergraph)**.** *The set of all minimal transversals of $\mathcal{H}$ forms the transversal hypergraph Tr($\mathcal{H}$).*

Example C.3. ([67]) The hypergraph $\mathcal{H} = \{\{v_1, v_2\}, \{v_1, v_2, v_3\}, \{v_2, v_3\}, \{v_3, v_4, v_5\}\}$ has the transversal hypergraph Tr($\mathcal{H}$) $= \{\{v_1, v_3\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_2, v_5\}\}$.

[1] https://en.wikipedia.org/wiki/Hypergraph



(a) A simple hypergraph **(b)** Two types of representation of a system [101]

Figure C.1.1: An introductory hypergraph example

Definition C.4 (Unions). *For simple hypergraph* $\mathcal{G} = \{g_1, g_2, ..., g_m\}$ *and* $\mathcal{H} = \{h_1, h_2, ..., h_{m'}\}$ *we have the following "union" operators*

$$\mathcal{G} \cup \mathcal{H} = \{g_1, g_2, ..., g_m, h_1, h_2, ..., h_{m'}\}$$

and

$$\mathcal{G} \vee \mathcal{H} = \{g_i \cup h_j : i = 1, 2, ..., m, j = 1, 2, ..., m'\}$$

An important property using these unions is the following.

Proposition C.1 ([20]). *Let* $\mathcal{G}$ *and* $\mathcal{H}$ *be simple hypergraphs. Then* $Tr(\mathcal{G} \cup \mathcal{H}) = min(Tr(\mathcal{G}) \vee Tr(\mathcal{H}))$.

Berge's algorithm generates (minimal) transversal hypergraphs using Proposition C.1 as follows. For a hypergraph $\mathcal{H} = \{e_1, e_2, ..., e_m\}$ let $\mathcal{H} = \{e_1, e_2, ..., e_i\}, i = 1, 2, ..., m$. We then have $\mathrm{Tr}(\mathcal{H}_i) = \min(\mathrm{Tr}(\mathcal{H}_{i-1}) \vee \mathrm{Tr}(e_i)) = \min(\mathrm{Tr}(\mathcal{H}_{i-1}) \vee \{\{v\} : v \in e_i\})$; and $\mathrm{Tr}(\mathcal{H}) = \mathrm{Tr}(\mathcal{H}_m)$. This implies a straightforward iterative computation process –the Berge-multiplication algorithm. A pseudocode listing is given in Algorithm C.1.1.

Algorithm C.1.1: Berge-multiplication algorithm

```
1 Tr(H) ← ∨{{v} : v ∈ e_1};
2 for i ← 2, ..., m do
3     Tr(H_i) ← min(Tr(H_{i-1}) ∨{{v} : v ∈ e_i});
4 return Tr(H_m);
```

Finally, we name the complements of transversal.

Definition C.5 ((Maximal) Independent Set). *Let* $\mathcal{H}$ *be a hypergraph. A subset of vertices of* $\mathcal{H}$ *is independent if it does not contain an edge. An independent set is maximal if no proper superset is independent.*

Note that the complement of an independent set in a hypergraph is a transversal. The complements of the maximal independent sets are the minimal transversals.

Applications of hypergraph have been deeply studied for two decades by Gallo et al. [60, 67]. Hypergraphs are also used to present biological, ecological and technological systems where the use of complex networks gives very limited information about the structure of the system [47]. Especially, applications of hypergraph into biological networks can draw on its own capability of modelling multiple relations of biological objects [101, 142]. Therefore, we see that the *directed hypergraph* is truly a useful and evident representation of metabolic networks.

C.2 Petri Net

A Petri net (PN) is graphically represented by a *directed bipartite graph* with two different types of nodes, called places $P = \{p_1, ..., p_n\}$ and transitions $T = \{t_1, ..., t_m\}$ [30, 64]. General speaking, places play the role of resources of the system, while transitions correspond to events that can change the state of the resources. *Places* (drawn as circles) classically model passive system elements such as conditions, states, or biological species (e.g. chemical compounds, for example, metabolites, proteins, complexes). *Transitions* (drawn as squares and rectangles) typically stand for active system elements such as events or chemical reactions (e.g. stoichiometric chemical reactions, complex formation, de-/phosphorylation). Weighted arcs (directed edges) connect places with transitions, depicting the relations between resources and events. The arcs of the graph are classified (with respect to transitions) as:

- *input* arcs: arrow-headed arcs from places (input places) to transitions.
- *output* arcs: arrow-headed arcs from transitions to places (output places).
- *inhibitor* arcs: circle-headed arcs from places to transitions.

A PN model can be formally defined in the following way [74]:

Definition C.6. *A PN model is an 5-tuple*

$$\mathcal{N} = \{P, T, F, W, M_0\}$$

where

- *P is the finite set of places;*
- *T is the finite set of transitions (the places P and transitions T are disjoint, $T \cap P = \emptyset$);*
- $F \subset (P \times T) \cup (T \times P)$ *is the flow relation;*
- $W : F \to (\mathbb{N} \backslash \{0\})$ *is the arc weight mapping;*
- $M_0 : P \to \mathbb{N}$ *is the initial marking representing the initial distribution of tokens. This is a function that associates with each place a natural number.*

Places hold zero or a positive number of *tokens*. The allocation of tokens over the places is called a *marking* (i.e. this process changes the state of the system). Formally, a marking is a function $M_0 : P \to \mathbb{N}$, and we refer to $M(p)$ for $p \in P$, as the number of tokens in place p in marking M. When modelling a system based on PN, we often describe the specification of an *initial marking* in its definition, which allocates a number of tokens to each place.

C.2.0.1 System Dynamics

So far we have dealt with the static component of a PN model. We now turn our attention to the dynamic evolution of the PN marking that is governed by transition firings which destroy and create tokens.

C.2.0.2 Enabling and firing rules

A transition is enabled if its input places contain at least the required numbers of tokens defined by the weight assigned to the arcs. Informally, we can say that the enabling rule defines the conditions that allow a transition t to fire, and the firing rule specifies the change of state produced by the transition.

Definition C.7 (Enabling). *Transition t is enabled in marking M if and only if*

- $\forall p \in {}^\bullet t, M(p) \geq O(t,p)$ ***and***
- $\forall p \in t^\bullet, M(p) \leq O(t,p)$

Definition C.8 (Firing). *The firing of transition t, enabled in marking M if and only if*

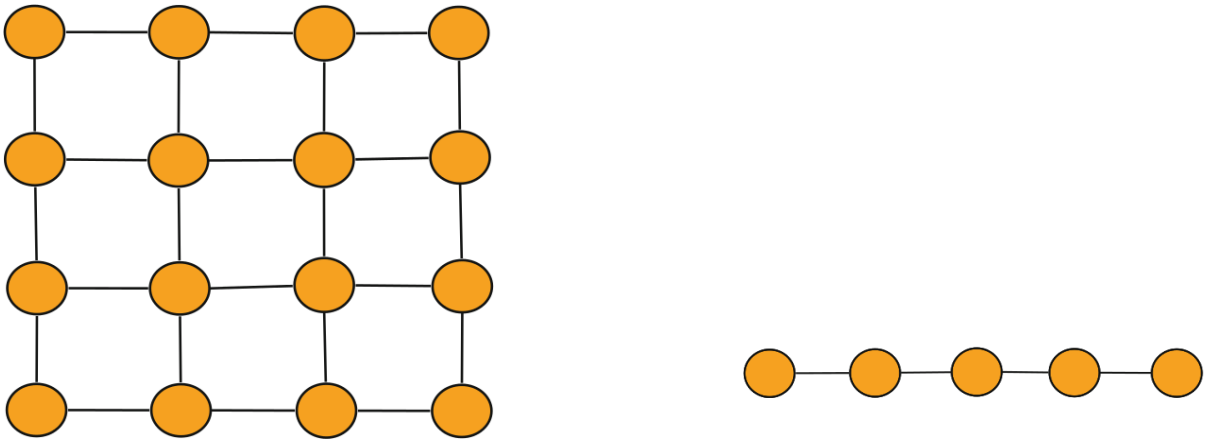
- $\forall p \in {}^\bullet t, M(p) \geq O(t,p)$ ***and***
- $\forall p \in t^\bullet, M(p) \leq O(t,p)$

The incidence matrix C of a PN with n places and m transitions is an $(n \times m)$ matrix, where every entry c_{ij} gives the token change on the place p_i by the firing of the transition t_j. Thus, read arcs are not reflected in the incidence matrix. The matrix C corresponds to the stoichiometric matrix in a metabolic network.

A *trap* is a set of places, whose all output transitions are also output of that set. That means, if a trap is marked with tokens, it will not be token-empty. A trap is maximal if it is not a proper subnet of any other trap [150].

C.2.1 Simple Networks

A simple network consists of regular connections among the vertices [43]. One of the most well-known examples therefore is the two-dimensional lattice (also called *grid graph*) as shown in Figure C.2.2. Here each vertex has the same number of neighbours. In particular, every pair of distinct nodes is connected by a unique edge, referred to *complete graph*. Despite its simplicity, such networks have been used extensively, e.g. in physics to study phenomena like controlling magnetism on surfaces [111]. Other examples of this class are linear chains or non rectangular lattices as used, e.g. in the context of protein structure prediction to model protein folding [2, 78].



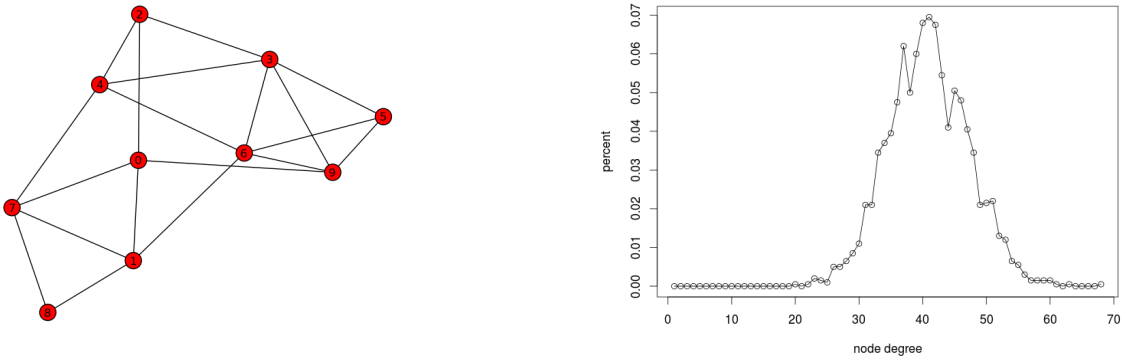
(a) Regular two-dimensional lattice.

(b) Linear, regular chain.

Figure C.2.2: Examples of simple networks [43]

C.2.2 Random Networks

Random networks studied extensively by Rapoport [143–145] and developed independently by Erdős and Rényi [44, 45]. In the paper in 1959 [44], Erdős and Rényi introduced the first model to generate random graphs consisting of n vertices and m edges. Starting with n disconnected vertices, the network is constructed by the addition of m edges at random, avoiding multiple and self connections. Then another similar model defines a random graph with n nodes that obtained by connecting every pair of nodes with probability P. The latter *random graph* model is often referred to the Erdős and Rényi (ER) model which can be considered the most basic one of complex networks. For this network model, $P(k)$ (the degree distribution, see Section 2.1.2) is a Poisson distribution (Figure C.2.3b).



(a) An example of a random graph.

(b) An average degree distribution.

Figure C.2.3: The random graph of Erdős and Rényi (ER) model. (a) an example of a random graph and (b) average degree distribution over 10 random networks formed by $2,000$ vertices using a probability $p = 0.2$.

Since 1959 to the 90 years of the 18^{th} century, almost all complex networks have been modelled and simulated randomly. The random network theory has widespread influenced on the growth many general sciences as well as rapidly developed of computer science. At the principle of the random network, each node in the given system has a few of links for connecting to its neighbours. Indeed, in a random network the degree vertex follows a Poisson distribution (also called bell shaped distribution - see the left chart in Figure 2.3). It means that it is extremely rare to find nodes that have significantly more or fewer links than the average. In other words, the probability that a node is connected to k other sites decreases exponentially for large k.

C.3 Minimum cut algorithms in Graph Theory

C.3.1 Flow-based approaches

The first approach for finding all minimum cuts of a graph is based on the maximum flow problem. The well-known max-flow [42, 53] theorem implies that a minimum $s-t$ cut can be found by computing the maximum flow between s and t.

C.3.1.1 Network flows

For a pair of vertices u, v, we define the *distance* $d_G(u,v)$ from u to v in G to be the minimal number of edges on the path from u to v in G. In the case that there is no such path, we define $d_G(u,v) = \infty$. A graph $G = (V, E)$ is a *flow network* if it has two distinguished vertices, a *source s* and a *sink t*, and a positive real number *capacity* $c(u,v)$ for each edge $(u,v) \in E$. We extend the capacity function to all vertex pairs by defining $c(u,v) = 0$ if $(u,v) \notin E$. A *flow* $f : E \to R$ on G is a real valued function on vertex pairs satisfying the following constraints:

$$f(u,v) \leqslant c(u,v), \forall (u,v) \in E \qquad \text{(Capacity constraint)} \qquad \text{(C.3.1)}$$

$$f(u,v) = -f(v,u), \forall (u,v) \in E \qquad \text{(Anti-symmetry constraint)} \qquad \text{(C.3.2)}$$

$$\sum f(u,v) = 0, \forall v \in V - \{s,t\} \qquad \text{(Flow conservation constraint)} \qquad \text{(C.3.3)}$$

The first condition says that the flow on a directed edge is never more than the capacity of that edge. The second says that flow on an edge is anti-symmetric: a units of flow on (u,v) implies $-a$ units of flow on (v,u). The final condition says that flow is conserved everywhere but the source and sink: the flow into each vertex is the same as the flow out of it. The *value* of a flow is the net flow into the sink, i.e.,

$$|f| = \sum_{v \in V} f(v,t)$$

The *maximum flow problem* is to determine a flow f for which $|f|$ is maximum. The well-known maxflow-mincut theorem [42, 53] states that in any network, the value of the maximum $s-t$ flow equals to the capacity of the $s-t$ minimum cut. An $s-t$ maximum flow algorithm can thus be used to find a $s-t$ minimum cut, and minimizes over all $\binom{2}{n}$ possible choices of s and t to yield a minimum cut.

C.3.1.2 Finding all minimum cuts in undirected weighted graphs

In 1961, Gomory and Hu [62] introduced a typical tree structure that can be able to find all minimum $s-t$ cuts for all $\binom{2}{n}$ pairs of s and t in an undirected and weighted graph. They showed that the number of distinct cuts in the graph is at most $n-1$ (rather than the naïve $\binom{2}{n}$). Furthermore, there is an efficient tree structure that can be maintained to compute this set of distinct cuts using only $n-1$ maximum flow computations. Given an undirected weighted graph $G=(V,E)$ with a capacity function c, a cut tree $T=(V,F)$ can be built from G is the tree having the same set of vertices V and the edge set F with a capacity function c' satisfying the following properties:

1. *Equivalent flow tree*: the smallest capacity of the edges on the path between s and t in T.

2. *Cut property*: a minimum cut $C_{s,t}$ is also a minimum cut in G.

The algorithm maintains a partition of V, $(S_1, S_2, ..., S_t)$ and a spanning tree T on the vertex set $\{S_1, S_2, ..., S_t\}$. Let w' be the function assigning weights to the edges of T. Initially, there exists only set $S_1 = V$. On each iteration, T satisfies the following invariant, that is, for any edge (S_i, S_j) in T, there are vertices a and b in S_i and S_j respectively such that $w'(S_i, S_j) = f(a, b)$ and the cut defined by edge (S_i, S_j) is a minimal $a-b$ cut in G. At the start, the algorithm chooses two nodes and calculates the minimal cut between them and the min cut groups. These groups are being separating into two graphs and the algorithm saves the minimal cut. Now at each of iteration the algorithm chooses two nodes from the same group and calculates the minimal cut between them, taking in account the other groups as a single point, which the maximal flow to and from it is the maximal flow that was found in one of the previous iterations. At the end of the algorithm **Gomory - Hu (GH) tree** is built. That tree represents the maximal flow between any two vertices in the graph, which is the minimal edge capacity of the path between those to edges. Following the demonstration in Appendix C.3.1.3 to understand steps for building a `GH` tree.

The complexity of building a `GH` tree depends on the technique used for the implementation of the algorithm. All the algorithms currently known for constructing a `GH` tree use $n-1$ minimal $s-t$ cut computations. In other words, any max-flow based approach for constructing a `GH` tree would have a running time of $(n-1)\times$ (time for computing a max-flow). In 1998, Karger and Levine [92] devised the algorithm to compute max-flow with the running time $\mathcal{O}(n^{2.16})$, so the best running time for building `GH` tree is $\mathcal{O}(n^{2.16}n)$. To the best of our knowledge, the current fastest $\mathcal{O}(mn)$ running time for `GH` tree construction on simple unweighted graphs with m edges and n vertices [71].

We have used `LEMON` open package, which realized `GH` algorithm, for testing on our own graphs built on real data.

C.3.1.3 Gomory-Hu demonstration

Initially, the algorithm starts with a trivial partition V and proceeds in $n-1$ iterations. The initial partition is the set $V = \{a, b, c, d, e, f\}$. Then the algorithm performs $n-1$ split operations as following:

- In each such split operation it chooses a set S_i with $|S_i| \geq 2$ and splits this set into two non-empty parts S and Y.
- S_i is then removed from T and replaced by X and Y.
- X and Y are connected by an edge, the edges that before the split incident to S_i are attached to either X or Y.

In the end this process gives a tree on the vertex set V. Details of the split operation can be described as below:

- Select a set S_i in the partition such that it contains at least two nodes. Let u and v be two distinct vertices of S_i.
- Compute the connected components of the forest obtained from the current tree T after deleting S_i. Each of these components corresponds to a set of vertices from V.
- Consider the graph H obtained from G by contracting these connected components into single nodes.
- Compute a minimum $u - v$ cut in H. Let A and B denote the two sides of this cut.
- Split S_i in T into two sets/nodes $S_i^u := S_i \cap A$ and $S_i^v := S_i \cap B$ and add edge $\{S_i^u, S_i^v\}$ with capacity $f_H(u, v)$.
- Replace and edge $\{S_i, S_x\}$ by $\{S_i^u, S_x\}$ if $S_x \subset A$ and by $\{S_i^v, S_x\}$ if $S_x \subset B$

Now we are seeing the way to construct an GH tree via the example explained in [62]. For testing, one can have a look GH algorithm to be implemented in LEMON[2] library. First, we start the algorithm with the original vertex set V to be denoted the initial partition. Figures C.3.4 to C.3.10 show the iterations performed during of building GH tree.

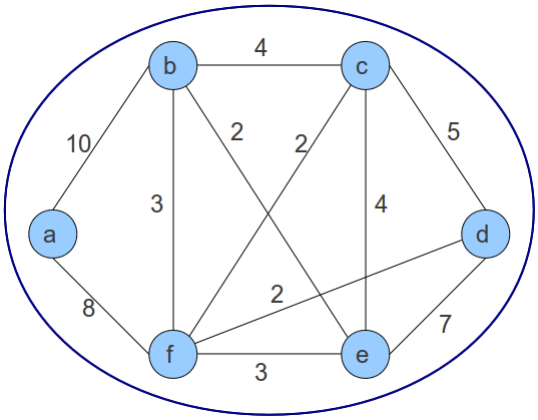
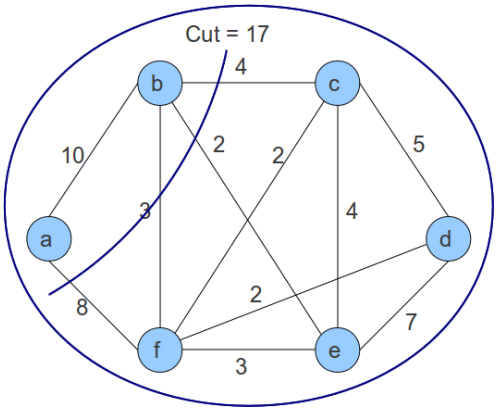
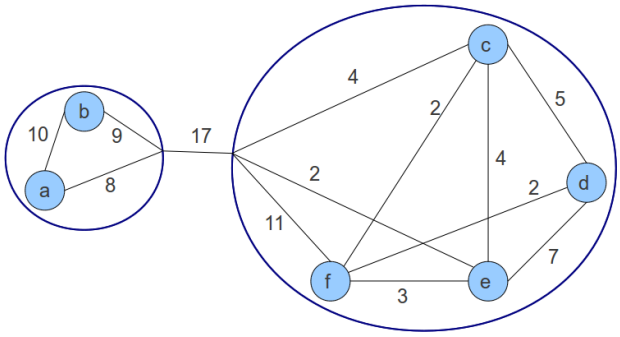


Figure C.3.4: Gomory-Hu Algorithm: Initial Step

[2]http://lemon.cs.elte.hu/trac/lemon

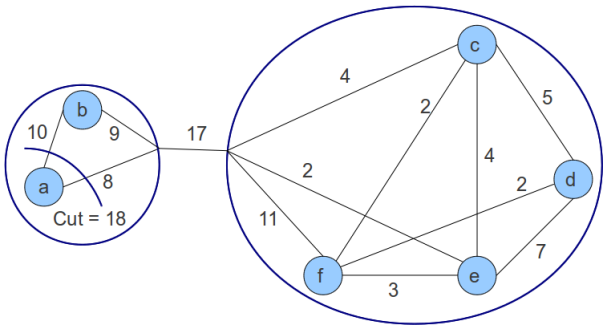


(a) Select b and f

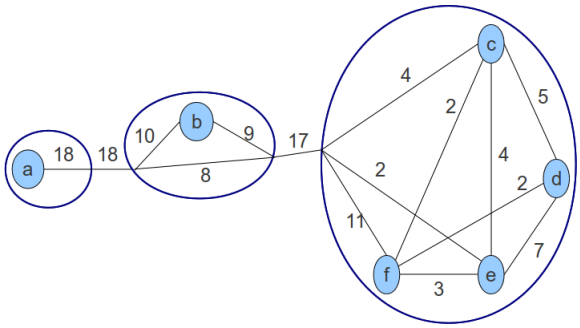


(b) $Partition_1 = (\{a, b\}, \{c, d, e, f\})$

Figure C.3.5: Gomory-Hu Algorithm: Iteration 1



(a) Select a and b

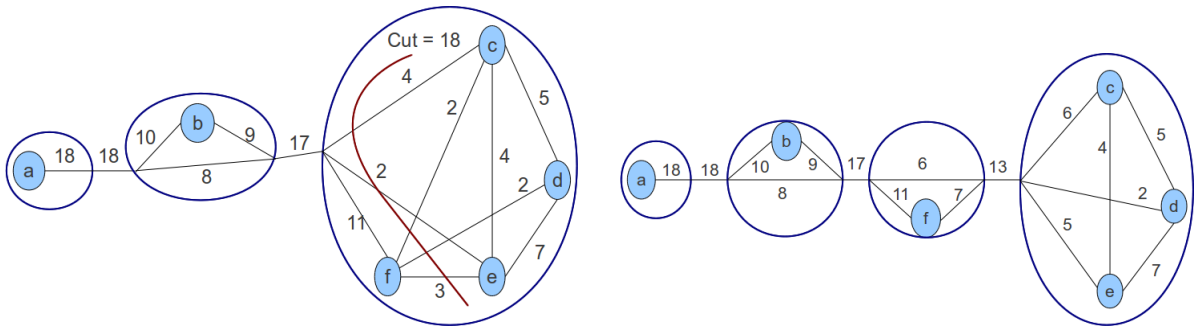


(b) $Partition_2 = (\{a\}, \{b\}, \{c, d, e, f\})$

Figure C.3.6: Gomory-Hu Algorithm: Iteration 2

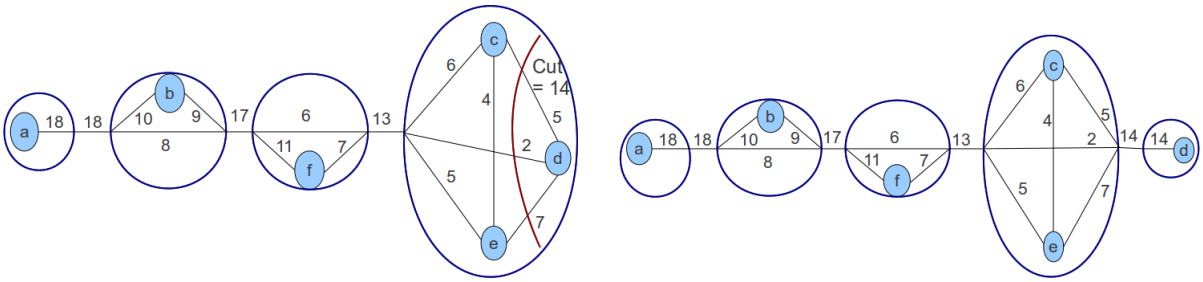
C.3.1.4 Determining a minimum cut in directed weighted graphs

A natural question to arise from GH algorithm is whether some of the information computed in one maximum flow computation can be reused in the next one. Hao and Orlin [68] (HO) answered this question in the affirmative. The key new idea is to use a push-relabel maximum flow algorithm to implement GH, and use the preflow and distance labelling from the last max-flow computation as a starting point for the current one.



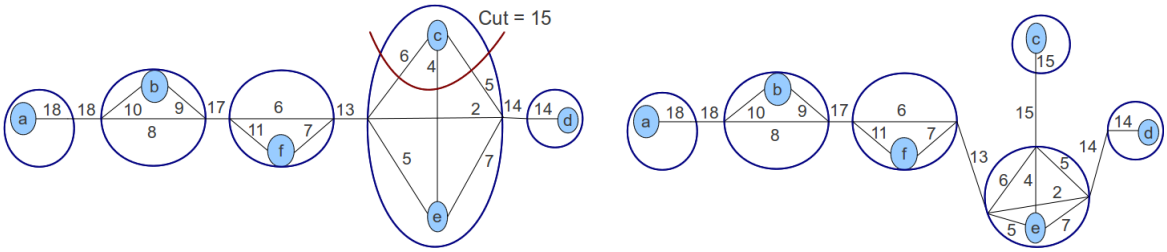
(a) Select c and f **(b)** $Partition_3 = (\{a\}, \{b\}, \{c, d, e\}, \{f\})$

Figure C.3.7: Gomory-Hu Algorithm: Iteration 3



(a) Select d and e **(b)** $Partition_4 = (\{a\}, \{b\}, \{c, e\}, \{d\}, \{f\})$

Figure C.3.8: Gomory-Hu Algorithm: Iteration 4



(a) Select c and e **(b)** $Partition_5 = (\{a\}, \{b\}, \{c\}, \{e\}, \{d\}, \{f\})$

Figure C.3.9: Gomory-Hu Algorithm: Iteration 5

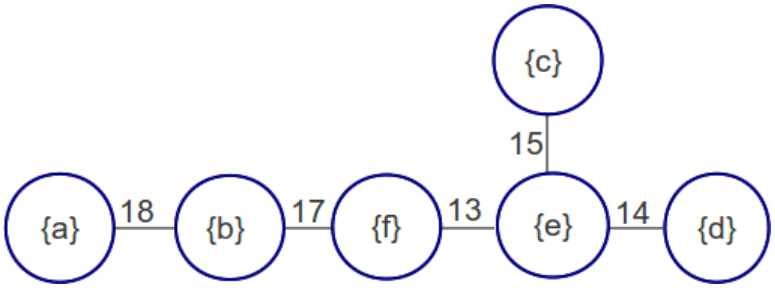


Figure C.3.10: Final Gomory-Hu Tree

They consider the problem of finding the minimum capacity cut in a directed network G with n nodes. One can use a maximum flow problem to find a minimum cut separating a designated source node s from a designated sink node t, and by varying the sink node one can find a minimum cut in G

as a sequence of at most $2n-2$ maximum flow problems. They then show how to reduce the running time of these $2n-2$ maximum flow algorithms to the running time for solving a single maximum flow problem. The resulting running time is $\mathcal{O}(mnlog(n^2/m))$ for finding the minimum cut in either a directed or an undirected network.

Algorithm C.3.1: Hao-Orlin's algorithm

```
1  λ ← ∞;
2  designate some vertex s, give it label 2n − 1, and saturate all of its outgoing arcs;
3  while there are non-source vertices do
4      read current;
5      if there are no awake vertices, awaken the top sleeping layer then
6          pick the awake vertex with minimum distance label as t;
7          PushRelabel(G, s, t) (always using GapRelabel, not Relabel);
8      if the excess at t is less than λ then
9          λ ← excess at t;
10     designate t a source vertex, and saturate all of its outgoing edges;
11 return λ;
```

It is not hard to check that the distance labels remain valid throughout the computation, which implies the correctness of the algorithm. Likewise, as in the maximum flow context, the distance labels are $\mathcal{O}(n)$ and only increase. It follows that using highest label selection, the time bound for HO is $\mathcal{O}(n^2\sqrt{m})$. The proof for FIFO selection with dynamic trees also carries over, giving a time bound of $\mathcal{O}(mnlog(n^2/m))$.

This algorithm is realised in LEMON library for finding a minimum cut in a directed graph $G = (A, N)$. It is a modified preflow push-relabel algorithm. The algorithm takes a fixed node $source \in N$ and consists of two phases: in the first phase it determines a minimum cut with $source$ on the source-side (i.e. a set $X \subsetneq V$ with $source \in X$ and minimum outgoing capacity) and in the second phase it determines a minimum cut with $source$ on the sink-side (i.e., a set $X \subsetneq V$ with $source \notin X$ and minimum outgoing capacity).

Other algorithms

Recently, several authors have studied efficient algorithms to enumerate all cut sets of a graph. By using the dual maximum flow problem, Curet [36] constructs a binary relation associated with an optimal maximum flow such that all minimum cost `s-t` are identified through the set of closures for this relation. The key improvement in Curet's approach is the use of graph theoretic techniques to rapidly enumerate the closures set.

C.3.2 Contraction Based Approaches

The new approach of the minimum cut problem is to repeatedly identify and *contract* edges that are not in the minimum cut until the minimum cut are obtained. It uses no flow-based techniques at all. This way can be only applied to undirected graphs, but they may be weighted.

C.3.2.1 Contract operation

Informally speaking, this operation takes an edge e with endpoints x and y and then contracts it into a new single node v_e which becomes adjacent to all former neighbours of x and y. The contraction of an edge makes the two nodes joined by e overlap, reducing the total number of nodes of the graph by one. Given a graph G and edge $(v, w) \in E$, we define $G/v, w$, the contraction of edge (v, w), by deleting w and replacing each edge of the form (w, x) by an edge (v, x). If this process creates parallel edges, we merge them and add the capacities. We also delete any self-loops. The steps are summarised in Algorithm C.3.2.

Algorithm C.3.2: GenericContractCut(G)

```
1 λ ← ∞;
2 while G has more than one node do
3 |   Either:;
4 |   1. identify an edge {v, w} that is not in some minimum cut;
5 |   2. compute λ_{v,w}(G) for some v and w and set λ = min{λ_{v,w}(G), λ};
6 |   G ← G/v, w;
7 return λ;
```

C.3.2.2 The first deterministic minimum cut algorithm

Nagamochi and Ibaraki [119] (`NI`) published the first deterministic minimum cut algorithm that is not based on a flow algorithm, has the slightly better running time of $\mathcal{O}(|V||E| + |V|^2 log|V|)$, but is still rather complicated. They gave a procedure called *scan-first search* that identifies and contracts an edge in $\mathcal{O}(|E| + |V|log|V|)$ time. This yields an algorithm that computes the minimum cut in $\mathcal{O}(|V||E| + |V|^2 log|V|)$.

This algorithm is realised in the LEMON library for undirected graphs.

C.3.2.3 The simple deterministic minimum cut algorithm

Stoer and Wagner [164] (`SW`) gave a simplified version of the Nagamochi and Ibaraki algorithm with the same running time. This simplification was subsequently discovered independently by Frank [55]. They proposed the following method for finding a minimum cut set of a graph G. A cut (S, T) of G is said to be a *global min-cut* if and only if the weight $w(S, T)$ of the cut is the smallest possible, i.e., for every other cut (S', T') of G we have $w(S, T) <= w(S', T')$. An $s - t$ min-cut is defined similarly. The algorithm is based on a theorem. Let s and t be two vertices of graph $G = (V, E)$. Let $G/s, t$ be

the graph obtained by contracting s and t. Then, a minimum cut of G can be obtained by taking the smaller of minimum $s-t$ cut and minimum cut of $G/s,t$. By this theorem, Stoer and Wagner comes up an algorithm. In each iteration of the algorithm, get two vertices which have a minimum cut to separate them, then contract these two vertices. This algorithm follows the theorem stated above. If the minimum cut is not current $s-t$ cut, then it should be in the graph $G/s,t$. The following is the pseudocode of the min-cut step in SW algorithm.

Algorithm C.3.3: MinCutPhase(G, w, a) in Stoer and Wagner algorithm

```
1  A ← {a};
2  while (A! = V) do
3      x ←Most Tightly Connected Vertex;
4      if (|A| == |V| − 2) then
5          s ← x;
6      if (|A| == |V| − 1) then
7          t ← x;
8      CurrentCut ← (A, Ā);
9      A ← x;
10     Contract(s, t);
11 return CurrentCut;
```

In Algorithm C.3.3, the vertex x is the *most tightly connected vertex* if it satisfies the following condition: $x \notin A : w(A,x) = \max\{w(A,y) \mid y \notin A\}$. Using the min-cut step in Algorithm C.3.3, Stoer-Wagner algorithm can be described as follow:

Algorithm C.3.4: Pseudocode of MinCut(G, w, a) in Stoer-Wagner algorithm

```
1  while (|V| > 1) do
2      CurrentCut ← MinCutPhase(G, w, a);
3      if (w(CurrentCut) < w(MinimumCut)) then
4          MinimumCut ← CurrentCut;
5  return MinimumCut;
```

C.3.2.4 The fastest known minimum cut randomized algorithm

This is a very nice randomized algorithm due to Karger and Stein (KS) that can compute the global minimum cut in near linear time with high probability. The idea of the algorithm [93] is based on the concept of contraction of an edge e in a graph $G = (V,E)$. The algorithm based on a sequences of contractions of a randomly chosen edge in a graph. The edges are selected proportional to its weight. The algorithm is recursive. One level of recursion consists of two independent trials of contraction of G to $\lceil n/\sqrt{2}+1 \rceil$ vertices followed by a recursion call.

Other algorithms

Sharafat and Márouzi [160] enhanced the recursive contraction algorithm. They modified the method proposed by Tsukiyama [171] by using the concept of iterative contraction Karger [91] and BFS ordering of vertices to develop a novel recursive contraction algorithm for scanning (enumerating and listing) all minimal cut sets of a given graph. Also, the authors introduced the concepts of pivot vertex, absorbable and unabsorbable clusters, and used them to develop an enhanced recursive contraction algorithm.

C.3.3 How to find all minimum cuts

The problem of finding all minimum cuts plays a critical importance in the design of real world complex systems. If a few of the links are cut or otherwise fail, the network may still be able to transmit messages between any pair of its nodes. If enough links fail, however, there will be at least one pair of nodes that cannot communicate with each other. Thus an important measure of the reliability of a network is the minimum number of links that must fail in order for this to happen. This number is referred to as the edge connectivity of the network and can be found by assigning a weight of 1 to each link and finding a minimum weight cut. In other applications, such as the open pit mining problem, we seek a minimum weight cut such that a specific pair of nodes, say node s and node t, are not in the same set. Solving this type of problem, known as a minimum $s-t$ cut problem, is a fundamental part of the calculations used to find the baseball elimination and clinch numbers. From the algorithms presented in literature, we can basically suggest some solutions to determine all minimum cuts in a certain graph as follows.

C.3.3.1 Basing on the definition of minimum cuts

Any graph has a finite number of cuts, so one could find the minimal cuts by enumerating and comparing the size of all the cuts basing on its definition. This is not a practical approach for large graphs which arise in real world applications since the number of cuts in such phenomena grows exponentially with the number of nodes.

C.3.3.2 Using ring sum of basic cut sets

A basic cut set is a cut set that contains only one edge and be independent. It can be considered as one of the approaches to reach the minimal cut set problem.

Ring sum Given two graph $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, then the ring sum of two graphs G_1 and G_2, denoted by $G_1 \bigoplus G_2$, is a graph which has $(V_1 \cup V_2, (E_1 \cup E_2) - (E_1 \cap E_2))$. The edges of a ring sum consist of edges which are either in G_1 or G_2, but which are not in both graphs. Ring sum is both commutative and associative. For example:

$$\{d,e,f\} \bigoplus \{f,g,h\} = \{d,e,g,h\}, \quad \text{(another cut set)} \tag{C.3.4}$$

$$\{d,e,g,h\} \bigoplus \{f,g,k\} = \{d,e,f,h,k\} \tag{C.3.5}$$

$$= \{d,e,f\} \cup \{h,k\}, \text{(an edge disjoint union of cut sets)} \tag{C.3.6}$$

Ariyoshi [11] defined a cut set graph with respect to a given graph G such that each edge of the graph corresponds to a pair of basic branch cut sets in the relation that the ring sum of these cut sets coincides with an incident branch cut set of G. A basic cut set can be generated by taking a ring sum of a number of incident cut sets. Deo [40] presented a method to be similar to the simple technique of finding a set of *fundamental circuits*. In the case of circuits, the other circuits in a graph can be created due to combinations of two or more fundamental circuits. Therefore, the term *fundamental cut set* is introduced in the correlation between the generating circuits and finding all minimal cut sets. A cut set S containing exactly one branch of a spanning tree T is called *fundamental cut set* with respect to T. Sometimes a fundamental cut set is also called a basic cut set. Every branch of a spanning tree defines a *unique* fundamental cut set. Using theorem 4.4 in [40], we have a method of generating additional cut sets from a number of given cut sets. Staring with two cut sets in a given graph, make a ring sum on them to have another one by this method. And the method is to use a vector spaces of a graph.

A Gaussian elimination method was presented by the author in [114]. By giving a suitable algebra for cut sets, it is possible to reduce the problem of enumerating all cut sets to the problem of solving a system of linear equations in this algebra. This method seems quite similar to the way Deo [40] applied to generate all cut sets. The author proposed the method for both directed and undirected graph, but he made an experiment with undirected graph. We can use this point to apply for our problem.

The algorithm of [3, 12, 81, 171] based on a blocking mechanism to determine all minimal cut sets of an undirected graphs, handle non-planar graphs with **multiple source and sink nodes** based on basic minimal paths.

C.3.3.3 Using Gomory-Hu algorithm

Gomory-Hu algorithm, which will be discussed later in Appendix C.3.1.2, can allow us to find all minimum cut sets in undirected unweighted graph.

These approaches can be enhanced for finding all cut sets in a directed graph with or without weights. The former is to extend `Hao-Orlin` algorithm assigning the weights of all edges to be 1. Then at each step of choose a next vertex in processing, we have several options (e.g., we have more than one vertex can be choose). The latter is to apply Deo's theory and the authors as shown in the previous sections that construct an algebra $C = (S, +, -)$ consists of a set S with two binary operations, sum and multiplication.

C.4 Applications of MCSs

The problem of finding a minimum cut set of a graph appears in many application s, for example, in network reliability, circuit design, clustering and information retrieval. More detailed, in order to evaluate the reliability of a network, we have to analyse potential faults that every of a such case is a set of edges in a cut set. Another usual application is to treat image segmentation as a graph partitioning problem, that is, to break graph into segments, etc. More about algorithms and applications of cut problem as well as network flows are found in [5, 76].

C.4.1 Evaluation of system reliability

A physical system would be quite unusual (or perhaps poorly designed) if replacing a failed component by a functioning one caused the system to change from the success to the failed state. Thus, we restrict consideration to structure functions that are monotonically increasing in each input variable [187]. These structures are called *coherent* and can be expressed as *cut sets*. Physically, a cut set is a set of components whose functioning (failure) ensures the functioning (failure) of the system.

The cut set method is a powerful one for evaluating the reliability of a system based on two reasons: (a) Easily to program in computer to find our solutions fast and efficient for any general network, (b) cut sets relate to the modes of system failure and therefore identity the distinct and discrete ways in which a system may fail. Following this fashion, we can define *a cut set is a set of system components which, when failed, caused failure of the system* [23]. So, we have also defined *a minimal cut set is a set of system components which, when failed, causes failure of the system but when any one component of the set has not failed, does not cause system failure*. From the definition of minimal cut sets it is evident that all components of each cut must be identified.

In advanced techniques of failure analysis, to evaluate the fault tree and determine the failure path, it is necessary to find the various minimal cut sets of the tree [141]. For this field, a cut set is defined *a set of basic events that have to take place for the top event to occur*. A cut set is said to be minimal when each of the basic events in the set is necessary and whose combination is sufficient to cause the top event. Each minimal cut set is an independent path for the failure to occur.

C.4.2 Fault Trees

Fault Trees are non-recursive Boolean networks studied in reliability and risk assessment of industrial systems

C.4.3 The k-cut problem

Let $G = (V, E)$, a weight function $w : E \to N$, and an integer $k \in [2..|V|]$. The k-cut problem is to create a partition of V into k disjoint sets $F = \{C_1, C_2, ..., C_k\}$ such that the following

formula Equation (C.4.1) is minimised

$$\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \sum_{\substack{v_1 \in C_i \\ v_2 \in C_j}} w(\{v_1, v_2\}) \tag{C.4.1}$$

The k-cut problem is an NP-complete problem which consists of finding a partition of a graph into k balanced parts such that the number of cut edges is minimised.

C.4.4 Image Segmentation of Computer Vision

Image Segmentation is the process of dividing an image into parts that have a strong correlation with objects or areas of the real world. Figure C.4.11 gets depicts how to get round line of cow[3]. By cutting an image into several segments[4], we treat every part simply and dependently in computer vision [28, 46, 51, 162].

Figure C.4.11: Dynamic image segmentation using Graph Cuts. The images in the first column are two consecutive frames of a video sequence and their respective segmentation, with the first image showing the user segmentation seeds (which are used as soft constraints on the segmentation). In column 2, we observe the n-edge flows obtained corresponding to the MAP solution of the MRFs representing the two problems. It can be clearly seen that the flows corresponding to the segmentation are similar. The flows from the first segmentation were used for finding the segmentation for the second frame. The time taken for this procedure was much less compared to that taken for finding the flows from scratch.

[3] http://masters.donntu.edu.ua/2007/kita/pankova/library/angl.htm
[4] http://www.cis.upenn.edu/~jshi/GraphTutorial/

Appendix D

Other results

D.1 Genes rules defined in regEfmtool

Here only shows basic gene rules. Let us suppose without loss of generality that Ri and Rj are two certain reactions used in logical expressions. Following C. Jungreuthmayer et al. [87, 88], three basic rules are stated as following:

Ri = (!fRj) means that (1) the reaction Ri carries a EFM while the reaction Rj must not carry any EFM, and (2) contrast to the first condition, the reaction Ri must not carry any EFM, if the reaction Rj carries that EFM. This rule can be said formally:

$$S_f = \{E \in EFMs : (Ri \in E \wedge Rj \notin E) \vee (Ri \notin E \wedge Rj \in E)\} \tag{D.1.1}$$

Using logical operations, equation D.1.1 can be rewritten by:

$$S_f = \{E \in EFMs : Ri \oplus Rj\} \tag{D.1.2}$$

Ri = (!0Rj) can be stated formally:

$$S_0 = S_f \cup \{E \in EFMs : Ri \in E \wedge Rj \in E\} \tag{D.1.3}$$

rewritten by:

$$S_0 = \{E \in EFMs : Ri \vee Rj\} \tag{D.1.4}$$

Ri=(!1Rj) can be written formally:

$$S_1 = S_f \cup \{E \in EFMs : Ri \notin E \wedge Rj \notin E\} \tag{D.1.5}$$

rewritten by

$$S_1 = \{E \in EFMs : Ri \notin E \wedge Rj \notin E\} \tag{D.1.6}$$

Simple_1 attached in regEfmtool If we do not indicate any rule, the EFMs set obtained consists of the following ones:

R1 R3 R4 R5 R6r R9
R1 R4 R5 R7
R1 R5 R8r
R4 R7 R8r
R1 R2 R6r R8r
R1 R2 R4 R6r R7
R1 R2 R3 R4 R9
R3 R4 R6r R8r R9

Trying to add three base boolean rules as described above, this will generate a subset of EFMs including EFMs:

R4 = (!fR3)	R4 = (!0R3)	R4 = (!1R3)
R4 R7 R8r	R1 R2 R3 R4 R9	R1 R4 R5 R7
R1 R2 R4 R6r R7	R1 R2 R4 R6r R7	R1 R2 R6r R8r
R1 R4 R5 R7	R1 R3 R4 R5 R6r R9	R1 R2 R4 R6r R7
	R1 R4 R5 R7	R1 R5 R8r
	R4 R7 R8r	R4 R7 R8r
	R3 R4 R6r R8r R9	

Simple_2 attached in regEfmtool Similarly, we do not limit the result by any gene rules. The list of EFMS can be obtained from computing with regEfmtool as follow:

R2t R4t R5t R7r R9 R11
R2t R4t R6t R7r R9 R10
R2t R4t R7r R8
R3t R4t R6t R7r R9 R10
R1t R2t R4t R8
R1t R2t R4t R5t R9 R11
R1t R4t R6t R9 R10
R1t R4t R5t R7r R9 R11
R1t R4t R7r R8
R3t R4t R5t R9 R11
R3t R4t R8

R7r = (!fR9)

R1t R4t R6t R9 R10
R2t R4t R7r R8
R1t R2t R4t R5t R9 R11
R1t R4t R7r R8
R3t R4t R5t R9 R11

R7r = (!0R9)

R2t R4t R7r R8
R2t R4t R6t R7r R9 R10
R3t R4t R6t R7r R9 R10
R1t R2t R4t R5t R9 R11
R1t R4t R5t R7r R9 R11
R1t R4t R6t R9 R10
R1t R4t R7r R8
R3t R4t R5t R9 R11
R2t R4t R5t R7r R9 R11

R7r = (!1R9)

R1t R2t R4t R8
R2t R4t R7r R8
R1t R2t R4t R5t R9 R11
R1t R4t R6t R9 R10
R1t R4t R7r R8
R3t R4t R8
R3t R4t R5t R9 R11

Table D.1.1: Histogram of occurrences of the reactions in EFMs/MCSs responding to sub networks

Reactions	Pathway	Whole Net	Vac_c	Vac_f	Vac_g	Vac_m	Vac_s	Vala_out	Vasp_out	Vcw	Vdag	Vgl_out	Vss
Global number of EFMs/MCSs			1,170	833	1,245	728	415	1,847	1,638	415	1,794	754	415
ala_up	Cytosol	15,696	182	102	153	104	51	1	286	51	234	52	51
Glc_up	Cytosol	109,224	0	0	0	0	0	0	0	0	0	0	0
gln_up_Vgs	Cytosol	89,594	962	600	900	520	300	**1,742**	1,300	300	**1,300**	**598**	300
NRJ1	Cytosol	112,036	**1,170**	786	**1,179**	**676**	**393**	**1,742**	**1,586**	393	**1,794**	**754**	393
NRJ1b	Cytosol	15,646	0	12	18	0	6	0	0	6	0	0	6
Tg6p	Cytosol	63,888	**1,125**	**833**	**1,245**	**713**	**415**	**1,831**	**1,623**	**415**	**1,794**	**739**	415
Ttp	Cytosol	74,797	858	624	933	546	311	**1,404**	**1,248**	311	**1,404**	**559**	311
Vala	Cytosol	54,430	494	386	579	338	193	**1,846**	624	193	858	260	193
Vala_out	Cytosol	38,736	312	284	426	234	142	**1,847**	338	142	624	208	142
Vald	Cytosol	65,705	**1,170**	**833**	**1,245**	**728**	**415**	**1,846**	**1,638**	**415**	**1,794**	**754**	415
Vcw	Cytosol	11,609	90	0	0	56	0	142	126	**415**	138	58	0
Vfbp	Cytosol	50,766	**1,170**	**833**	**1,245**	**728**	**415**	**1,846**	**1,638**	**415**	**1,794**	**754**	415
Vg6pdh_p	Cytosol	74,696	481	344	516	260	172	871	715	172	897	299	172
Vg6pdh_Vepi_Tx5p	Cytosol	82,650	481	344	516	260	172	871	715	172	897	299	172
Vgapdh_Vpgk_Vpgm_Veno	Cytosol	79,378	858	624	933	546	311	**1,404**	**1,248**	311	**1,404**	**559**	311
Vgl_out	Cytosol	19,608	0	116	174	0	58	208	104	58	0	**754**	58
Vhk1	Cytosol	21,649	270	0	415	168	0	426	378	0	414	174	0
Vhk2	Cytosol	110,464	270	415	0	168	0	426	378	0	414	174	0
Vinv	Cytosol	39,153	540	**830**	**1,245**	336	0	852	756	0	828	348	0
Vpepc	Cytosol	61,665	624	412	618	312	206	598	**1,222**	206	650	**572**	206
Vpfk	Cytosol	31,576	90	0	0	56	0	142	126	0	138	58	0
Vpgi	Cytosol	53,630	**1,170**	**833**	**1,245**	**728**	**415**	**1,846**	**1,638**	**415**	**1,794**	**754**	415
Vpk	Cytosol	37,959	364	248	372	286	124	**1,066**	234	124	520	260	124
Vrbco	Cytosol	35,147	**1,170**	**833**	**1,245**	**728**	**415**	**1,846**	**1,638**	**415**	**1,794**	**754**	415
Vsps_Vspace	Cytosol	72,065	540	**833**	830	336	**415**	852	756	0	828	348	0
Vsusy	Cytosol	36,582	270	3	415	168	0	426	378	0	414	174	0
Vtpi	Cytosol	72,517	858	624	933	546	311	**1,404**	**1,248**	311	**1,404**	**559**	311
Vut_NRJ3_Vpglm	Cytosol	66,762	720	**830**	**1,245**	448	**415**	**1,136**	**1,008**	**415**	**1,104**	464	0
NRJ2_Vkgdh_Vsdh_Vfum	Mitochondria	50,017	572	338	507	442	169	702	832	169	234	0	169
Vaco_Vidh	Mitochondria	53,196	**390**	346	519	390	173	832	572	173	**1,066**	364	173
Vasp_out_Vasp	Mitochondria	37,671	260	252	378	156	126	338	**1,638**	126	624	104	126
Vcl	Mitochondria	35,166	90	164	246	272	82	718	462	82	**1,122**	58	82
Vcs	Mitochondria	43,527	810	242	363	200	121	238	366	121	138	394	121
Vgdh	Mitochondria	48,002	494	286	429	234	143	286	234	143	624	182	143
Vmdh	Mitochondria	52,820	520	358	537	286	179	**1,066**	338	179	832	260	179
Vme	Mitochondria	47,718	442	324	486	0	162	**1,014**	338	162	832	260	162
Vpdh	Mitochondria	60,693	780	370	555	156	185	104	520	185	832	364	185
Tpep	Plastid	67,936	858	624	933	546	311	**1,404**	**1,248**	311	**1,404**	**559**	311
Vat_Vss_Vpglm_p	Plastid	22,469	90	0	0	56	0	142	126	0	138	58	415
Vepi_p	Plastid	67,223	**1,116**	**833**	**1,245**	**710**	**415**	**1,828**	**1,620**	**415**	**1,794**	**736**	415
Vgapdh_p	Plastid	79,378	858	624	933	546	311	**1,404**	**1,248**	311	**1,404**	**559**	311
Vpfk_p_Vald_p_Vtpi_p	Plastid	42,859	90	0	0	56	0	142	126	0	138	58	0
Vpgi_p	Plastid	70,471	**1,080**	**833**	**1,245**	**698**	**415**	**1,816**	**1,608**	**415**	**1,794**	**724**	415
Vpk_p_Vpdh_p_VFAx_Vdag_Vglyc3P	Plastid	56,518	0	276	414	234	138	624	624	138	**1,794**	0	138
Vriso_p_Vtkx_p_Vtald_p	Plastid	96,612	**1,062**	**833**	**1,245**	**692**	**415**	**1,810**	**1,602**	**415**	**1,794**	**718**	415
Vac_c	Vacuole	28,054	**1,170**	180	270	0	90	312	260	90	0	0	90
Vac_f	Vacuole	34,752	180	**833**	415	112	0	284	252	0	276	116	0
Vac_g	Vacuole	1,246	270	415	**1,245**	168	0	426	378	0	414	174	0
Vac_m	Vacuole	19,428	0	112	168	**728**	56	234	156	56	234	0	56
Vac_s	Vacuole	19,392	90	0	0	56	**415**	142	126	0	138	58	0

D.2 Drawings corresponding the sub networks without the unused reactions

D.2.1 `Vac_f`, `Vac_g` and `Vac_s` in the `Vacuole` compartment

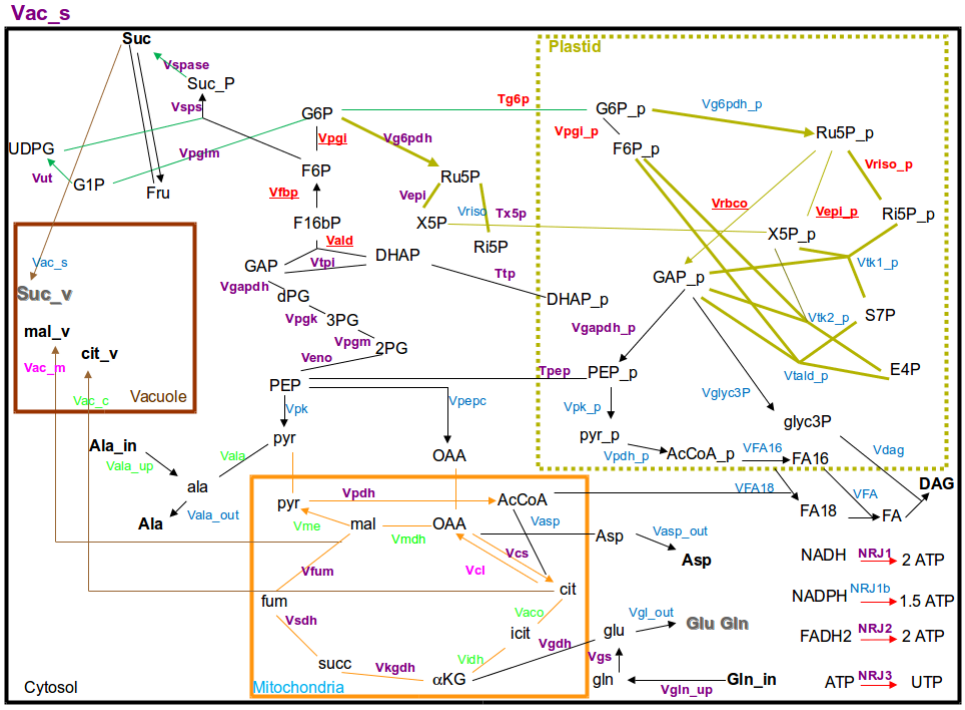


Figure D.2.1: Model of `Vac_s` after the removal of the unused reactions

D.2.2 `Vgl_out` in the `Cytosol` compartment

D.2.3 `Vss` in the `Plastid` compartment

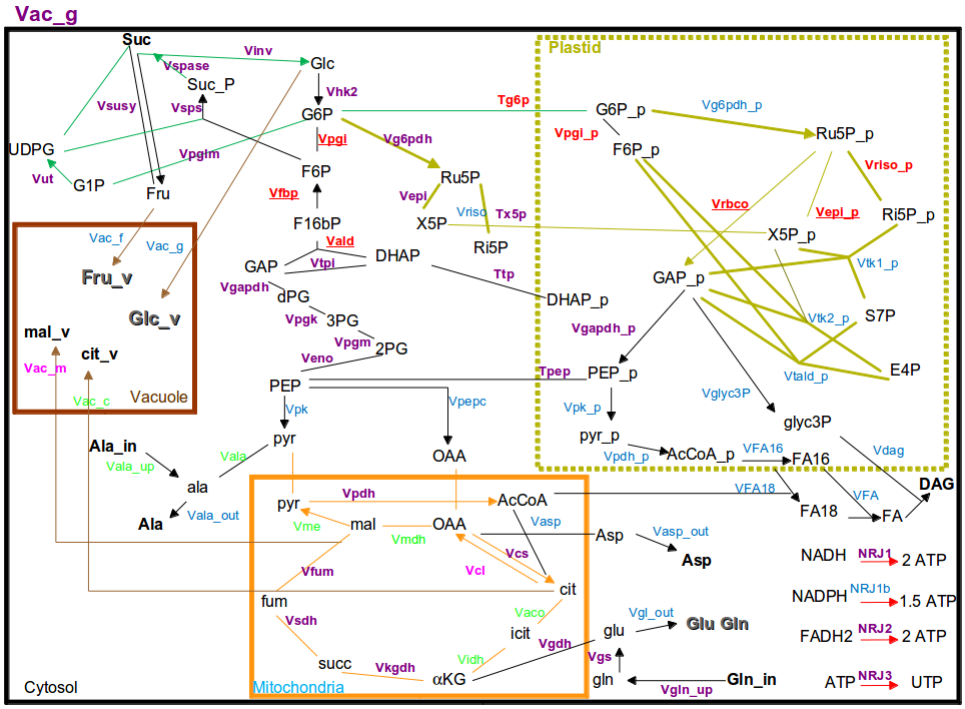


Figure D.2.2: Model of `Vac_g` after the removal of the unused reactions

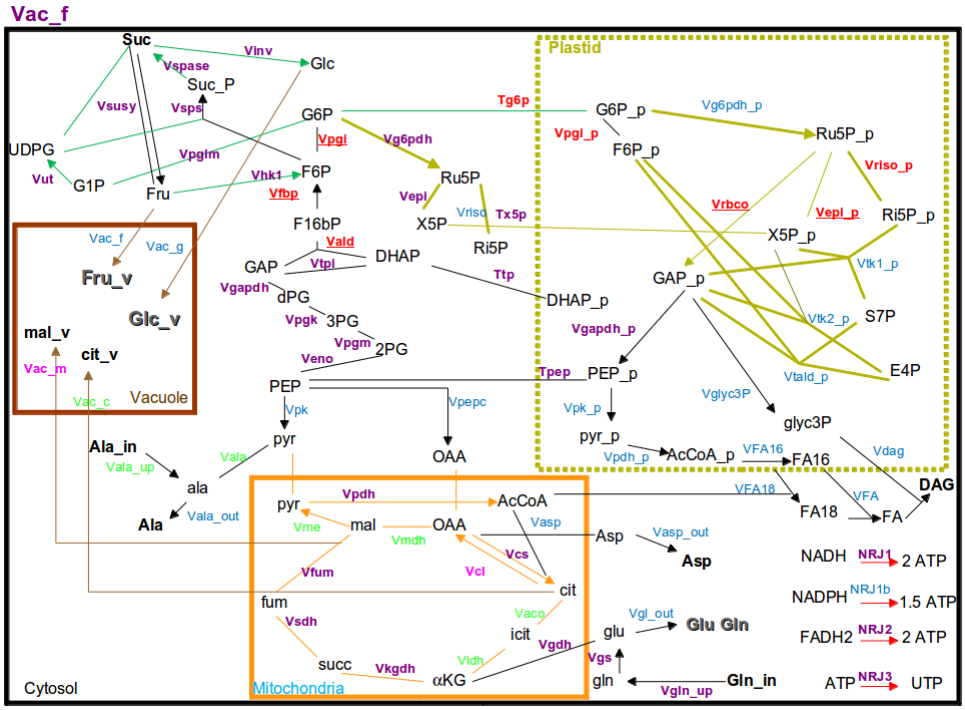


Figure D.2.3: Model of `Vac_f` after the removal of the unused reactions

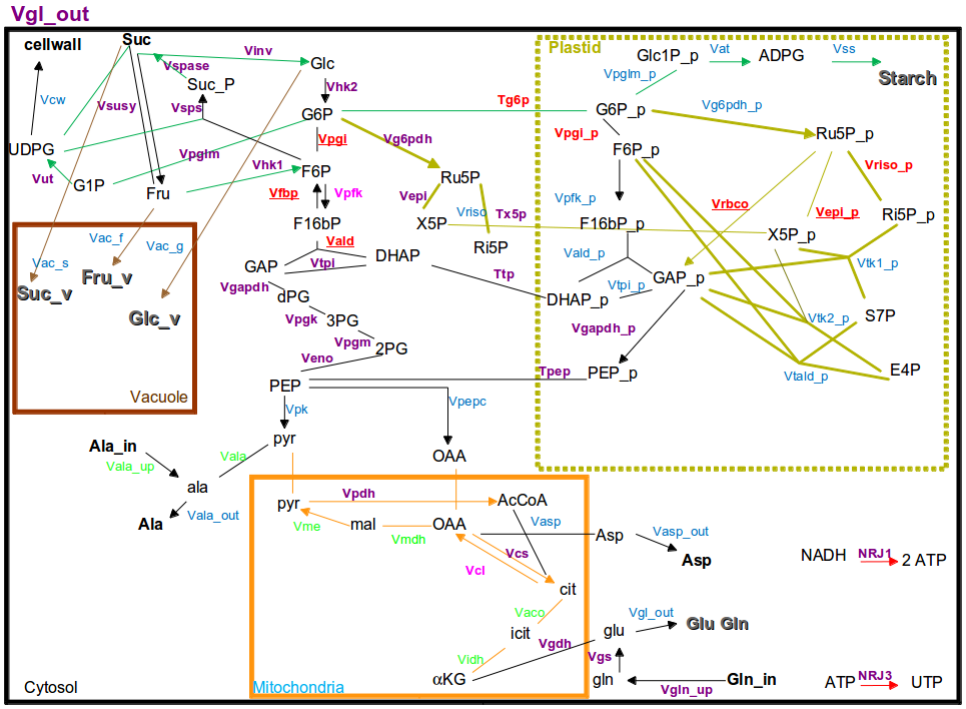


Figure D.2.4: Model of `Vgl_out` after the removal of the unused reactions

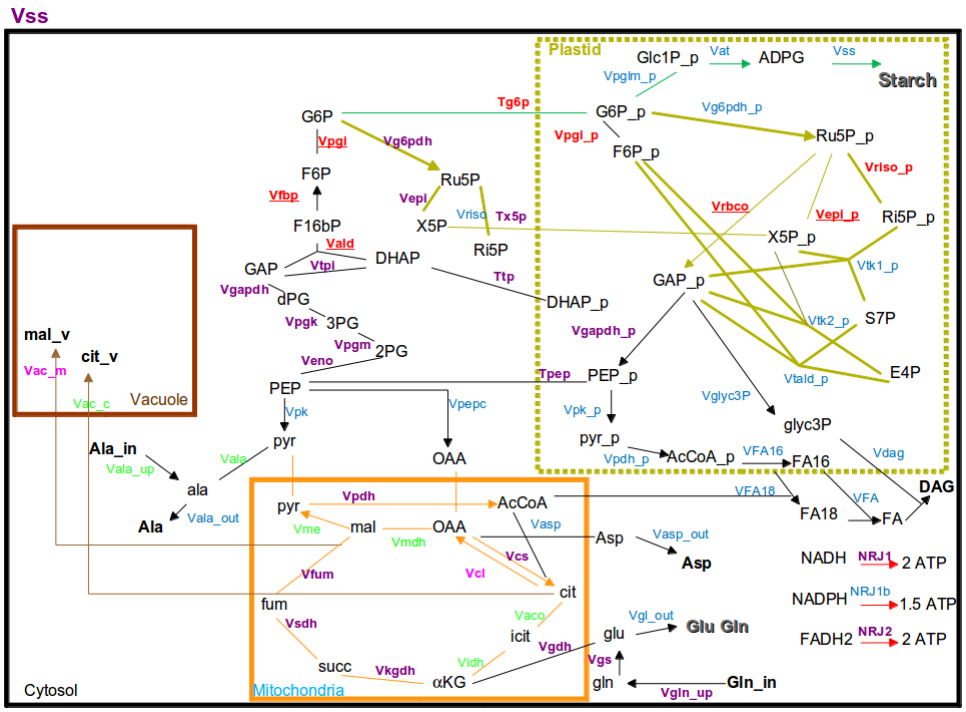


Figure D.2.5: Model of `Vss` after the removal of the unused reactions

D.3 List of all different motifs

➤ Ttp-Veno

1 Ttp || Tpep Vgapdh_Vpgk_Vpgm_Veno
2 Ttp || Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno
3 Ttp || Vgapdh_Vpgk_Vpgm_Veno
4 Ttp Vgdh || Vpk Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno
5 Ttp Vgdh || Vpk Vpepc Vpdh Vcs Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh
6 Vac_c Ttp Vgdh Vaco_Vidh || Vpk Vgapdh_p Vgl_out Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno
7 Vac_m Ttp || Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno
8 Vmdh Ttp || Vac_m Vgapdh_Vpgk_Vpgm_Veno
9 Vme Ttp Vgdh || Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno
10 Vme Vala_out Ttp Vala || Vmdh Vgapdh_Vpgk_Vpgm_Veno
11 Vpdh Vcs Vac_c Ttp || Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno
12 Vpdh Vcs Vac_c Vmdh Ttp Vgdh || Vala_out Vac_m Vala Vgapdh_Vpgk_Vpgm_Veno
13 Vpdh Vcs Vac_c Vmdh Ttp || Vme Vgapdh_Vpgk_Vpgm_Veno
14 Vpdh Vcs Vgl_out Ttp || Vala_out Vac_c Vala Vgapdh_Vpgk_Vpgm_Veno
15 Vpdh Vcs Vmdh Ttp Vaco_Vidh || Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno
16 Vpdh Vcs Vme Vac_c Vmdh Ttp || Vgapdh_p Vac_m Tpep Vgapdh_Vpgk_Vpgm_Veno
17 Vpepc Vac_c Ttp || Vmdh Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh
18 Vpepc Vac_m Ttp || Vmdh Vgdh Vgapdh_Vpgk_Vpgm_Veno
19 Vpepc Vala_out Ttp Vala || Vpdh Vcs Vac_c Vgdh Vgapdh_Vpgk_Vpgm_Veno
20 Vpepc Vme Vala_out Vmdh Ttp Vala || Vgdh Vgapdh_Vpgk_Vpgm_Veno
21 Vpepc Vpdh Vcs Vgl_out Ttp || Vala_out Vac_c Vala Vgapdh_Vpgk_Vpgm_Veno
22 Vpk Vac_c Ttp Vaco_Vidh || Vpepc Vme Vgl_out Vmdh Vgapdh_Vpgk_Vpgm_Veno
23 Vpk Vac_m Ttp || Vpepc Vme Vmdh Vgapdh_Vpgk_Vpgm_Veno
24 Vpk Vala_out Ttp Vala || Vpepc Vpdh Vcs Vac_c Vgdh Vgapdh_Vpgk_Vpgm_Veno
25 Vpk Vala_out Ttp Vala || Vpepc Vpdh Vcs Vme Vmdh Vgapdh_Vpgk_Vpgm_Veno Vaco_Vi

➤ Tpep-Vtpi

1 Tpep Vgapdh_Vpgk_Vpgm_Veno || Vtpi
2 Tpep || Vtpi
3 Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno || Vtpi
4 Vgapdh_p Tpep || Vtpi
5 Vgapdh_p Vac_c Tpep Vaco_Vidh || Vgl_out Vtpi
6 Vgapdh_p Vac_c Tpep Vgdh || Vpepc Vpdh Vcs Vme Vgl_out Vtpi Vmdh
7 Vgapdh_p Vac_c Tpep || Vgl_out Vtpi Vgdh Vaco_Vidh
8 Vgapdh_p Vac_m Tpep Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vme Vac_c Vtpi Vmdh
9 Vgapdh_p Vac_m Tpep || Vpdh Vcs Vme NRJ1 Vac_c Vtpi
10 Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vtpi Vmdh Vaco_Vidh
11 Vgapdh_p Vala_out Tpep Vala Vgdh || Vpepc Vpdh Vcs NRJ1 Vac_c Vtpi
12 Vgapdh_p Vala_out Tpep Vala || Vpdh Vcs NRJ1 Vgl_out Vtpi Vaco_Vidh
13 Vgapdh_p Vgl_out Tpep Vgdh || Vpepc Vme Vala_out Vac_c Vtpi Vmdh Vala Vaco_Vidh
14 Vgapdh_p Vmdh Tpep Vgdh || Vpepc Vpdh Vcs Vme Vac_c Vtpi
15 Vgapdh_p Vmdh Tpep || Vpepc Vtpi
16 Vgapdh_p Vme Vala_out Vmdh Tpep Vala || Vpk Vpdh Vcs Vtpi Vaco_Vidh

17 Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh || Vala_out Vtpi Vala
18 Vpdh Vcs Vgapdh_p Tpep Vgdh Vaco_Vidh || Vpepc Vala_out Vtpi Vala
19 Vpdh Vcs Vgapdh_p Vme NRJ1 Vac_c Tpep || Vac_m Vtpi
20 Vpepc Vgapdh_p Vme Tpep Vgdh || Vpk Vala_out Vtpi Vmdh Vala
21 Vpepc Vgapdh_p Vme Vala_out Tpep Vala || Vpk Vpdh Vcs Vac_c Vtpi Vgdh
22 Vpepc Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh || Vala_out Vtpi Vala
23 Vpepc Vpdh Vcs Vgapdh_p Vme Vac_c Tpep || Vpk Vala_out Vtpi Vmdh Vala
24 Vpepc Vpdh Vcs Vgapdh_p Vme Vgl_out Vmdh Tpep || Vpk Vala_out Vac_c Vtpi Vala
25 Vpk Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno || Vtpi Vgdh
26 Vpk Vgapdh_p Vgl_out Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno || Vac_c Vtpi Vgdh Vaco_Vidh
27 Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno || Vac_m Vtpi
28 Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno || Vme Vtpi Vgdh
29 Vpk Vpepc Vpdh Vcs Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh || Vtpi Vgdh

➤ Ttp-Vgapdh_p

1 Ttp || Vgapdh_p
2 Ttp || Vgapdh_p Tpep
3 Ttp || Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno
4 Ttp Vgdh || Vpepc Vgapdh_p
5 Ttp Vgdh || Vpk Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno
6 Ttp Vgdh || Vpk Vpepc Vpdh Vcs Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh
7 Vac_c Ttp Vgdh Vaco_Vidh || Vpk Vgapdh_p Vgl_out Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno
8 Vac_m Ttp || Vpdh Vcs Vgapdh_p Vme NRJ1 Vac_c Tpep
9 Vac_m Ttp || Vpdh Vgapdh_p Vme
10 Vac_m Ttp || Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno
11 Vala_out Ttp Vala || Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh
12 Vala_out Ttp Vala || Vpepc Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh
13 Vgl_out Ttp || Vgapdh_p Vac_c Tpep Vaco_Vidh
14 Vgl_out Ttp Vgdh Vaco_Vidh || Vgapdh_p Vac_c Tpep
15 Vme Ttp Vgdh || Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno
16 Vpdh Vcs NRJ1 Vgl_out Ttp Vaco_Vidh || Vgapdh_p Vala_out Tpep Vala
17 Vpdh Vcs Vmdh Ttp Vaco_Vidh || Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno
18 Vpdh Vcs Vme NRJ1 Vac_c Ttp || Vgapdh_p Vac_m Tpep
19 Vpdh Vcs Vme Vac_c Vmdh Ttp || Vgapdh_p Vac_m Tpep Vgapdh_Vpgk_Vpgm_Veno
20 Vpdh Vme Ttp || Vgapdh_p Vac_m
21 Vpepc Ttp || Vgapdh_p Vgdh
22 Vpepc Ttp || Vgapdh_p Vmdh Tpep
23 Vpepc Vala_out Ttp Vala || Vpdh Vcs Vgapdh_p Tpep Vgdh Vaco_Vidh
24 Vpepc Vme Vala_out Vac_c Vmdh Ttp Vala Vaco_Vidh || Vgapdh_p Vgl_out Tpep Vgdh
25 Vpepc Vpdh Vcs NRJ1 Vac_c Ttp || Vgapdh_p Vala_out Tpep Vala Vgdh
26 Vpepc Vpdh Vcs Vme Vac_c Ttp || Vgapdh_p Vmdh Tpep Vgdh
27 Vpepc Vpdh Vcs Vme Vgl_out Vmdh Ttp || Vgapdh_p Vac_c Tpep Vgdh
28 Vpk Vala_out Vac_c Ttp Vala || Vpepc Vpdh Vcs Vgapdh_p Vme Vgl_out Vmdh Tpep
29 Vpk Vala_out Vmdh Ttp Vala || Vpepc Vgapdh_p Vme Tpep Vgdh
30 Vpk Vala_out Vmdh Ttp Vala || Vpepc Vpdh Vcs Vgapdh_p Vme Vac_c Tpep
31 Vpk Vpdh Vcs Ttp Vaco_Vidh || Vgapdh_p Vme Vala_out Vmdh Tpep Vala

```
32 Vpk Vpdh Vcs Vac_c Ttp Vgdh || Vpepc Vgapdh_p Vme Vala_out Tpep Vala
```

➤ Tpep-Ttp

```
1 Tpep || Ttp
2 Tpep Vgapdh_Vpgk_Vpgm_Veno || Ttp
3 Vgapdh_p Tpep || Ttp
4 Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno || Ttp
5 Vgapdh_p Vac_c Tpep Vaco_Vidh || Vgl_out Ttp
6 Vgapdh_p Vac_c Tpep Vgdh || Vpepc Vpdh Vcs Vme Vgl_out Vmdh Ttp
7 Vgapdh_p Vac_c Tpep || Vgl_out Ttp Vgdh Vaco_Vidh
8 Vgapdh_p Vac_m Tpep Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vme Vac_c Vmdh Ttp
9 Vgapdh_p Vac_m Tpep || Vpdh Vcs Vme NRJ1 Vac_c Ttp
10 Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vmdh Ttp Vaco_Vidh
11 Vgapdh_p Vala_out Tpep Vala Vgdh || Vpepc Vpdh Vcs NRJ1 Vac_c Ttp
12 Vgapdh_p Vala_out Tpep Vala || Vpdh Vcs NRJ1 Vgl_out Ttp Vaco_Vidh
13 Vgapdh_p Vgl_out Tpep Vgdh || Vpepc Vme Vala_out Vac_c Vmdh Ttp Vala Vaco_Vidh
14 Vgapdh_p Vmdh Tpep Vgdh || Vpepc Vpdh Vcs Vme Vac_c Ttp
15 Vgapdh_p Vmdh Tpep || Vpepc Ttp
16 Vgapdh_p Vme Vala_out Vmdh Tpep Vala || Vpk Vpdh Vcs Ttp Vaco_Vidh
17 Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh || Vala_out Ttp Vala
18 Vpdh Vcs Vgapdh_p Tpep Vgdh Vaco_Vidh || Vpepc Vala_out Ttp Vala
19 Vpdh Vcs Vgapdh_p Vme NRJ1 Vac_c Tpep || Vac_m Ttp
20 Vpepc Vgapdh_p Vme Tpep Vgdh || Vpk Vala_out Vmdh Ttp Vala
21 Vpepc Vgapdh_p Vme Vala_out Tpep Vala || Vpk Vpdh Vcs Vac_c Ttp Vgdh
22 Vpepc Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh || Vala_out Ttp Vala
23 Vpepc Vpdh Vcs Vgapdh_p Vme Vac_c Tpep || Vpk Vala_out Vmdh Ttp Vala
24 Vpepc Vpdh Vcs Vgapdh_p Vme Vgl_out Vmdh Tpep || Vpk Vala_out Vac_c Ttp Vala
25 Vpk Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno || Ttp Vgdh
26 Vpk Vgapdh_p Vgl_out Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno || Vac_c Ttp Vgdh Vaco_Vidh
27 Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno || Vac_m Ttp
28 Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno || Vme Ttp Vgdh
29 Vpk Vpepc Vpdh Vcs Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh || Ttp Vgdh
```

➤ Ttp-Vtpi

```
1 Vtpi || Ttp
```

➤ Vtpi-Veno

```
1 Tpep Vgapdh_Vpgk_Vpgm_Veno || Vtpi
2 Vac_m Vgapdh_Vpgk_Vpgm_Veno || Vtpi Vmdh
3 Vala_out Vac_c Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vgl_out Vtpi
4 Vala_out Vac_c Vala Vgapdh_Vpgk_Vpgm_Veno || Vpepc Vpdh Vcs Vgl_out Vtpi
5 Vala_out Vac_m Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vac_c Vtpi Vmdh Vgdh
6 Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vac_c Vtpi
7 Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno || Vtpi
8 Vgapdh_p Vac_m Tpep Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vme Vac_c Vtpi Vmdh
9 Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vtpi Vmdh Vaco_Vidh
10 Vgapdh_Vpgk_Vpgm_Veno || Vtpi
```

```
11 Vgdh Vgapdh_Vpgk_Vpgm_Veno || Vpepc Vme Vala_out Vtpi Vmdh Vala
12 Vmdh Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh || Vpepc Vac_c Vtpi
13 Vmdh Vgapdh_Vpgk_Vpgm_Veno || Vme Vala_out Vtpi Vala
14 Vmdh Vgdh Vgapdh_Vpgk_Vpgm_Veno || Vpepc Vac_m Vtpi
15 Vme Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vac_c Vtpi Vmdh
16 Vpdh Vcs Vac_c Vgdh Vgapdh_Vpgk_Vpgm_Veno || Vpepc Vala_out Vtpi Vala
17 Vpepc Vme Vgl_out Vmdh Vgapdh_Vpgk_Vpgm_Veno || Vpk Vac_c Vtpi Vaco_Vidh
18 Vpepc Vme Vmdh Vgapdh_Vpgk_Vpgm_Veno || Vpk Vac_m Vtpi
19 Vpepc Vpdh Vcs Vac_c Vgdh Vgapdh_Vpgk_Vpgm_Veno || Vpk Vala_out Vtpi Vala
20 Vpepc Vpdh Vcs Vme Vac_c Vgapdh_Vpgk_Vpgm_Veno || Vpk Vala_out Vtpi Vmdh Vala
21 Vpepc Vpdh Vcs Vme Vmdh Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh || Vpk Vala_out Vtpi Vala
22 Vpk Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno || Vtpi Vgdh
23 Vpk Vgapdh_p Vgl_out Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno || Vac_c Vtpi Vgdh Vaco_Vidh
24 Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno || Vac_m Vtpi
25 Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno || Vme Vtpi Vgdh
26 Vpk Vpepc Vpdh Vcs Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh || Vtpi Vgdh
```

➤ Vtpi-Vgapdh_p

```
1 Vac_c Vtpi Vgdh Vaco_Vidh || Vpk Vgapdh_p Vgl_out Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno
2 Vac_m Vtpi || Vpdh Vcs Vgapdh_p Vme NRJ1 Vac_c Tpep
3 Vac_m Vtpi || Vpdh Vgapdh_p Vme
4 Vac_m Vtpi || Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno
5 Vala_out Vtpi Vala || Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh
6 Vala_out Vtpi Vala || Vpepc Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh
7 Vgl_out Vtpi || Vgapdh_p Vac_c Tpep Vaco_Vidh
8 Vgl_out Vtpi Vgdh Vaco_Vidh || Vgapdh_p Vac_c Tpep
9 Vme Vtpi Vgdh || Vpk Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgapdh_Vpgk_Vpgm_Veno
10 Vpdh Vcs NRJ1 Vgl_out Vtpi Vaco_Vidh || Vgapdh_p Vala_out Tpep Vala
11 Vpdh Vcs Vme NRJ1 Vac_c Vtpi || Vgapdh_p Vac_m Tpep
12 Vpdh Vcs Vme Vac_c Vtpi Vmdh || Vgapdh_p Vac_m Tpep Vgapdh_Vpgk_Vpgm_Veno
13 Vpdh Vcs Vtpi Vmdh Vaco_Vidh || Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno
14 Vpdh Vme Vtpi || Vgapdh_p Vac_m
15 Vpepc Vala_out Vtpi Vala || Vpdh Vcs Vgapdh_p Tpep Vgdh Vaco_Vidh
16 Vpepc Vme Vala_out Vac_c Vtpi Vmdh Vala Vaco_Vidh || Vgapdh_p Vgl_out Tpep Vgdh
17 Vpepc Vpdh Vcs NRJ1 Vac_c Vtpi || Vgapdh_p Vala_out Tpep Vala Vgdh
18 Vpepc Vpdh Vcs Vme Vac_c Vtpi || Vgapdh_p Vmdh Tpep Vgdh
19 Vpepc Vpdh Vcs Vme Vgl_out Vtpi Vmdh || Vgapdh_p Vac_c Tpep Vgdh
20 Vpepc Vtpi || Vgapdh_p Vgdh
21 Vpepc Vtpi || Vgapdh_p Vmdh Tpep
22 Vpk Vala_out Vac_c Vtpi Vala || Vpepc Vpdh Vcs Vgapdh_p Vme Vgl_out Vmdh Tpep
23 Vpk Vala_out Vtpi Vmdh Vala || Vpepc Vgapdh_p Vme Tpep Vgdh
24 Vpk Vala_out Vtpi Vmdh Vala || Vpepc Vpdh Vcs Vgapdh_p Vme Vac_c Tpep
25 Vpk Vpdh Vcs Vac_c Vtpi Vgdh || Vpepc Vgapdh_p Vme Vala_out Tpep Vala
26 Vpk Vpdh Vcs Vtpi Vaco_Vidh || Vgapdh_p Vme Vala_out Vmdh Tpep Vala
27 Vtpi || Vgapdh_p
28 Vtpi || Vgapdh_p Tpep
```

29 Vtpi || Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno
30 Vtpi Vgdh || Vpepc Vgapdh_p
31 Vtpi Vgdh || Vpk Vgapdh_p Vala_out Tpep Vala Vgapdh_Vpgk_Vpgm_Veno
32 Vtpi Vgdh || Vpk Vpepc Vpdh Vcs Vgapdh_p Tpep Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh

➤ Veno-Vgapdh_p

1 Tpep Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p
2 Vac_c Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh || Vgapdh_p Vgl_out Tpep
3 Vac_m Tpep Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vgapdh_p Vme
4 Vac_m Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vmdh Tpep
5 Vac_m Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vgapdh_p Vme NRJ1 Vac_c Tpep
6 Vala_out Vac_m Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgdh
7 Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh
8 Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vgapdh_p Tpep Vaco_Vidh
9 Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vgapdh_p Vac_c Tpep
10 Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno || Vpepc Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh
11 Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno || Vpepc Vpdh Vcs Vgapdh_p Tpep Vaco_Vidh
12 Vala_out Vmdh Vala Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vgapdh_p Vac_c Tpep Vgdh
13 Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p
14 Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Tpep
15 Vgdh Vgapdh_Vpgk_Vpgm_Veno || Vpepc Vgapdh_p
16 Vgl_out Vgdh Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh || Vgapdh_p Vac_c Tpep
17 Vmdh Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vme Vala_out Tpep Vala
18 Vme Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep
19 Vpdh Vcs NRJ1 Vgl_out Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh || Vgapdh_p Vala_out Tpep Vala
20 Vpdh Vcs Vac_c Vgdh Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vala_out Vmdh Tpep Vala
21 Vpdh Vcs Vac_c Vmdh Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vme Tpep
22 Vpdh Vcs Vac_c Vmdh Vgdh Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vala_out Vac_m Tpep Vala
23 Vpdh Vcs Vgl_out Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vala_out Vac_c Tpep Vala
24 Vpdh Vcs Vme NRJ1 Vac_c Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vac_m Tpep
25 Vpdh Vcs Vme Vac_c Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vac_m Tpep
26 Vpdh Vme Tpep Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vac_m
27 Vpepc Tpep Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vgdh
28 Vpepc Vpdh Vcs NRJ1 Vac_c Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vala_out Tpep Vala Vgdh
29 Vpepc Vpdh Vcs Vgl_out Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vala_out Vac_c Tpep Vala

➤ Tpep-Vgapdh_p

1 Tpep || Vgapdh_p
2 Tpep Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p
3 Tpep Vgdh || Vpepc Vgapdh_p
4 Vac_m Tpep Vgapdh_Vpgk_Vpgm_Veno || Vpdh Vgapdh_p Vme
5 Vpdh Vme Tpep Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vac_m
6 Vpepc Tpep Vgapdh_Vpgk_Vpgm_Veno || Vgapdh_p Vgdh

➤ Tpep-Veno

1 Tpep || Vgapdh_Vpgk_Vpgm_Veno
2 Vgapdh_p Tpep || Vgapdh_Vpgk_Vpgm_Veno

3 Vgapdh_p Vac_c Tpep || Vgl_out Vgdh Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh
4 Vgapdh_p Vac_m Tpep || Vpdh Vcs Vme NRJ1 Vac_c Vgapdh_Vpgk_Vpgm_Veno
5 Vgapdh_p Vac_m Tpep || Vpdh Vcs Vme Vac_c Vgapdh_Vpgk_Vpgm_Veno
6 Vgapdh_p Vala_out Tpep Vala Vgdh || Vpepc Vpdh Vcs NRJ1 Vac_c Vgapdh_Vpgk_Vpgm_Veno
7 Vgapdh_p Vala_out Tpep Vala || Vpdh Vcs NRJ1 Vgl_out Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh
8 Vgapdh_p Vala_out Vac_c Tpep Vala || Vpdh Vcs Vgl_out Vgapdh_Vpgk_Vpgm_Veno
9 Vgapdh_p Vala_out Vac_c Tpep Vala || Vpepc Vpdh Vcs Vgl_out Vgapdh_Vpgk_Vpgm_Veno
10 Vgapdh_p Vala_out Vac_m Tpep Vala || Vpdh Vcs Vac_c Vmdh Vgdh Vgapdh_Vpgk_Vpgm_Veno
11 Vgapdh_p Vala_out Vmdh Tpep Vala || Vpdh Vcs Vac_c Vgdh Vgapdh_Vpgk_Vpgm_Veno
12 Vgapdh_p Vgl_out Tpep || Vac_c Vgapdh_Vpgk_Vpgm_Veno Vaco_Vidh
13 Vgapdh_p Vmdh Tpep || Vac_m Vgapdh_Vpgk_Vpgm_Veno
14 Vgapdh_p Vme Tpep || Vpdh Vcs Vac_c Vmdh Vgapdh_Vpgk_Vpgm_Veno
15 Vgapdh_p Vme Vala_out Tpep Vala || Vmdh Vgapdh_Vpgk_Vpgm_Veno
16 Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh || Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno
17 Vpdh Vcs Vgapdh_p Tpep Vaco_Vidh || Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno
18 Vpdh Vcs Vgapdh_p Vac_c Tpep || Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno
19 Vpdh Vcs Vgapdh_p Vac_c Tpep Vgdh || Vala_out Vmdh Vala Vgapdh_Vpgk_Vpgm_Veno
20 Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep Vgdh || Vala_out Vac_m Vala Vgapdh_Vpgk_Vpgm_Veno
21 Vpdh Vcs Vgapdh_p Vac_c Vmdh Tpep || Vme Vgapdh_Vpgk_Vpgm_Veno
22 Vpdh Vcs Vgapdh_p Vme NRJ1 Vac_c Tpep || Vac_m Vgapdh_Vpgk_Vpgm_Veno
23 Vpepc Vpdh Vcs Vgapdh_p NRJ1 Vgl_out Tpep Vaco_Vidh || Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno
24 Vpepc Vpdh Vcs Vgapdh_p Tpep Vaco_Vidh || Vala_out Vala Vgapdh_Vpgk_Vpgm_Veno

Appendix E

Extending works

The results have just presented in the last section explained a part of the MNHPC organisation. However, there has remained functions can be studied and compared to these 5 functions. Thus we have tried doing the analysis on 6 case studies more in MNHPC.

In this thesis, we have chosen 6 metabolism functions (aka. at all 11 are Vac_c, Vac_f, Vac_g, Vac_m, Vac_s, Vala_out, Vasp_out, Vcw, Vdag, Vgl_out and Vss) that are main products of metabolic processes. These functions play the important role of input substances and output products in metabolisms. It is interesting to figure out the relevance of them and to find differences from in vivo designs to vitro studies. One of the first steps would be to construct an overall statistics as displayed in Table E.0.1.

Table E.0.1: Topological properties of 11 biological functions. Nb. (1): Number of EFMs with Glc_up; Nb. (2): Number of MCSs computing directly from its EFMs matrix; Nb. (3): Number of MCSs containing Glc_up using grep -w

No.	Items/Functions	Nb. EFMs	Nb. (1)	Nb. (2)	Nb. (3)
1	Whole network	114,614	109,224	93,009	7,956
2	Vac_c	28,054	26,884	3,984	103
3	Vac_f	34,752	33,919	14,445	15
4	Vac_g	1,246	1	553	552
5	Vac_m	19,428	18,700	4,251	95
6	Vac_s	19,392	18,977	14,456	1,621
7	Vala_out	38,736	36,889	3,958	99
8	Vasp_out	37,671	36,033	5,989	99
9	Vcw	11,609	11,194	16,151	1,871
10	Vdag	56,518	54,724	5,014	17
11	Vgl_out	19,608	18,854	5,499	87
12	Vss	22,469	22,054	13,886	1621

For all sub-matrix, **the number of EFMs is always bigger** than the **number of MCSs**, moreover **the average length of EFMs** is also **bigger** than **the average size of MCSs**. That confirms the

hypothesis that **MCSs could be easier to analyse than EFMs**.

E.1 Finding the isolated reactions

It's interesting to find the list of the reactions which are always absent in all metabolic pathways. Without loss of generality, we can eliminate those reactions out the network. In other words, the removal of them can reduce the complexity of our network. From the occurrence distribution, the zero values will be removed.

E.2 Finding the longest chain of reactions

Opposite to the list of isolated reactions, we have tried to find the longest series of reactions participating in all metabolic pathways. This point can be explained how the network till works even if `Glc_up` is removed out of all the processes.

E.3 Clustering the reactions into groups

Based on the discrete data computed in the previous steps, we divided 11 functions into 5 groups as follows: presents the differences among groups based on the distributions of reaction occurrences in the EFMs and MCSs set.

E.3.0.1 Vac_s, Vcw and Vss

This group has the identical numbers of EFMs (e.g. 415) but their contents are incompletely analogous. So how to analyse the results? Fortunately, we found 5 reactions which the values are different.

In the three cases, we have deliberately withdraw reactions which the occur values are different. There are 5 reactions have to remove out the EFM lists. They are Vac_s, Vat_Vss_Vpglm_p, Vcw, Vsps_Vspace and Vut_NRJ3_Vpglm. The way to do this step is as follows:

- In the EFMs list of Vac_s, removing Vac_s certainly, Vsps_Vspace and Vut_NRJ3_Vpglm.
- In the EFMs list of Vcw, removing Vcw certainly and Vsps_Vspace.
- In the EFMs list of Vss, removing Vat_Vss_Vpglm_p only.

and the final lists of the three functions are completely identical. In theory, it has an existence of the interrelationship among the MCSs sets. But reality, the frequency distributions of reactions of MCSs do not have any similar features.

The occurred values of MCSs (computed by the method [5] in the computational models with the values $1,621$, $1,871$ and $1,621$ corresponding to `Vac_s`, `Vcw` and `Vss` respectively) are a little bit different. The occurrence distributions of the reactions of the EFMs set in these case studies are completely similar. However, the ones of the MCS sets do not have the same trend. The similarities seem to suggest a question of the existence of a relationship among the candidate EFMs in three case studies. Consequently, we chose MCSs with the smallest sizes such as 2, 3, 4, even if the long sizes. Moreover, the smaller frequencies can give us helpful information.

Take into account the MCSs values computed using The method used [2] in the computational models, they are the same (equals 550). Only 4/50 reactions (means lines) have the same value with the exception of 16/50 zero lines. It should recall that these MCSs do not completely contain any Glc_up because they are directly computed from the EFMs matrices without Glc_up at the initial.

Summary up, the group of three reactions has the same set of the feasible routes to generate them.

E.3.0.2 Vac_m, Vala_out and Vasp_out

Contrast to the first case study, the frequency distributions of the reactions in the MCSs set are likely similar (95, 99 and 99 of Vac_m, Vala_out and Vasp_out respectively) while the ones of the EFMs are distinct from each others. Some values of Vac_m differ from Vala_out and Vasp_out. Only ala_up is in Vala_out not in the two others. The values of Vala_out and Vasp_out are almost identical except ala_up. Thus we verify the probability of two sets consistent if ala_up removes out the MCSs. Using the method [5] in the computational model, the results indicated in the following are small differences between

Vala_out and Vasp_out

```
< Glc_up Vmdh Vriso_p_Vtkx_p_Vtald_p
< Glc_up Vmdh Vpgi_p
---
> Glc_up Vme Vriso_p_Vtkx_p_Vtald_p
> Glc_up Vme Vpgi_p
---
< Glc_up Vepi_p Vmdh
> Glc_up Vepi_p Vme
---
< Glc_up Tg6p Vmdh
> Glc_up Tg6p Vme
```

Similarly, we tried to compare Vala_out with Vac_m and Vasp_out with Vac_m. The differences are not so much.

Vac_m and Vala_out

```
> Glc_up Vme Vriso_p_Vtkx_p_Vtald_p
> Glc_up Vme Vpgi_p
---
> Glc_up Vepi_p Vme
---
> Glc_up Tg6p Vme
```

Vac_m and Vasp_out

```
> Glc_up Vmdh Vriso_p_Vtkx_p_Vtald_p
> Glc_up Vmdh Vpgi_p
---
```

```
> Glc_up Vepi_p Vmdh
---
> Glc_up Tg6p Vmdh
```

E.3.0.3 Vala_out, Vasp_out and Vgl_out

The results of MCS computation from grepping MCSs containing Glc_up (e.g. the approach [5] in computational model) depict the almost similar between Vala_out (99) and Vasp_out (99) product. Looking at the list of MCSs, we found that removing ala_up out the MCSs might decrease differences because it only appears in Vala_out. The distinction at the moment reduces significantly.

Vala_out and Vasp_out

```
< Glc_up Vme Vriso_p_Vtkx_p_Vtald_p
< Glc_up Vme Vpgi_p
> Glc_up Vmdh Vriso_p_Vtkx_p_Vtald_p
> Glc_up Vmdh Vpgi_p
---
< Glc_up Vepi_p Vme
> Glc_up Vepi_p Vmdh
---
< Glc_up Tg6p Vme
> Glc_up Tg6p Vmdh
```

Vala_out and Vgl_out

Vasp_out and Vgl_out

E.3.0.4 Vac_c and Vac_m

In this case, MCSs distributions seem to be similar whereas the ones of EFMs are different.

E.3.0.5 Vac_f and Vac_g

Following the method [2] in the computational models, the number of MCSs (552) are the same.

E.3.0.6 Vac_f and Vdag

Using the method [5] in the computational models In this case, MCSs distributions seem to be similar whereas the ones of EFMs are different. The belows show the differences of the two functions.

```
> Glc_up Vasp_out_Vasp Vat_Vss_Vpglm_p Vhk2 Vpfk_p_Vald_p_Vtpi_p Vpk
Vpk_p_Vpdh_p_VFAx_Vdag_Vglyc3P
> Glc_up Vasp_out_Vasp Vat_Vss_Vpglm_p Vhk2 Vpdh Vpfk_p_Vald_p_Vtpi_p
Vpk_p_Vpdh_p_VFAx_Vdag_Vglyc3P
---
< Glc_up Vald Vg6pdh_p
> Glc_up Vac_f Vasp_out_Vasp Vhk2 Vinv Vme Vpfk_p_Vald_p_Vtpi_p Vpk
Vpk_p_Vpdh_p_VFAx_Vdag_Vglyc3P
> Glc_up Vac_f Vasp_out_Vasp Vhk2 Vinv Vme Vpdh Vpfk_p_Vald_p_Vtpi_p
```

```
Vpk_p_Vpdh_p_VFAx_Vdag_Vglyc3P
--
< Glc_up Tg6p Vg6pdh_p
```

E.3.1 Finding motifs

Based on the above analyses, we found the assembled reactions that participating in most pathways. They are reactions appearing into the functions owns similar characteristics. First of all, two computational models (numbering [2] and [5]) were chosen to find common minimal cut sets.

E.3.1.1 MCSs with size 2

Model 2:

Model 5: The functions Vac_m, Vac_s, Vala_out, Vcw and Vss do not any MCSs with size 2. After merging all MCSs size 2 of the remain functions, the list of MCSs size 2 shared is {Vac_c, Vaco_Vidh, Vald, Vepi_p, Vfbp, Vgapdh_p, Vmdh, Vpfk, Vpgi, Vpgi_p, Vrbco}.

E.3.1.2 MCSs with size 3

Model 2:

Model 5:

E.3.2 Analysis of Minimal Cut Sets

From Appendix B.4, we computed the different cases and associated with 5 case studies E.3 to verify our hypothesis. To keep two computational models, the results show some interesting points in EFMs set. The steps of the verificaton can be described as follows:

- Each of groups, Vac_c and Vac_m for instance, select the reactions which costs equal to the maximum values. That means they always appear in all EFMs. In our first attempt with the group (Vac_c, Vac_m), the reactions Vpgi, Vald, Vfbp are selected as a motif. To facilitate in the next steps, named three reactions like **S**.

- Finding candidates to add to the motif **S**. To find it, basing on the MCSs with size 2, 3, 4 etc. For example, we discover Tg6p and Ttp often appear in MCSs.

- Determining the number of MCSs and finding out the smallest cases. For instance, here **S** + Ttp has 5 cases with small values.

- Computing frequency distribution of the reactions.

- There are 2 groups as mentioned in the previous sections are approved in this example.

Verification of trivial MCSs in the computational results A question raising in our mind that whether the trivial MCSs appear in the final result? For example, MCSs with only one reaction that is also the objective function. Using TCA cycle graph, we choose all EFMs containing T6. There is 13 EFMs containing T6. Clearly, T6 is the reaction appearing all 13 EFMs. It is one of the trivial MCSs has to be in the final set of MCSs. However, we did not find the ones like that in the previous experiments. So using CNA and mcsCalculator to compute MCSs and the results obtained have some strange notes.

McsCalculator has a line to enable compressed mode. Compiling mcsCalculator twice with turn on/off this line to get two its versions: compressed and uncompressed. If using mcsCalculator for fpcwithoutSubEnzymes, the program gives the same results in compressed and uncompressed mode.

Appendix F

Scientific Activities

F.1 Publications

Books and book chapters

Beurton-Aimar M., **Nguyen Vu-Ngoc T.**, Sophié C. Metabolic Network Reconstruction and Their Topological Analysis. In: Dieuaide-Noubhani M, Alonso AP, eds. *Plant Metabolic Flux Analysis: Methods and Protocols*. 1st ed. Humana Press; 2014:35-64. Available at: http://goo.gl/ryWpkg.

Proceedings

- **Nguyen Vu Ngoc T.**, Beurton-Aimar M., Sophié C. Minimal Cut Sets and Its Application to Study Metabolic Pathway Structures. In: Patrick Amar, François Ké ès, Vic Norris, eds. *Proceedings of the Nice Spring School on Advances in Systems and Synthetic Biology*. Nice, France; 2013:71-81.

- **Nguyen Vu Ngoc T.**, Beurton-Aimar M., Colombié S. Graph for Biology: Managing graph in metabolism modeling context. In: Vietnam National Conference on Information Technology and Telecommunications 14th.; 2011.

F.2 Abstracts, Posters and Presentations

- **Nguyen Vu Ngoc Tung**, Beurton-Aimar Marie and Colombié Sophie. Topological Analysis of Metabolic Networks : Application to Heterotrophic Plant Cell Network (Abstract & Poster). *European Conference on Complex Systems 2014* (ECCS'14) (22–26 September 2014 at IMT, Lucca, Italy). Available at: http://goo.gl/bpi5aN.

- **Nguyen Vu Ngoc Tung**, Beurton-Aimar Marie and Colombié Sophie. Using Topological Analysis to Study Metabolism of Heterotrophic Plant Cell Network (Abstract & Poster). *BioNetVisA workshop: From biological network reconstruction to data visualization and analysis in molecular biology and medicine* (September 7–10, 2014 at Strasbourg, France). Available at: http://goo.gl/OsYF2R.

- **Nguyen Vu Ngoc Tung**, Beurton-Aimar Marie and Colombié Sophie. Heterotrophic Plant Cell Network Analysis: Comparison Between EFMs and MCSs Methods (Abstract & Poster).

Metabolic Pathway Analysis 2013 Conference. (16-20 September 2013 Corpus Christi College, Oxford, UK). Available at: http://goo.gl/upSxHw.

- M. Beurton-Aimar, **N. T. Nguyen-Vu**, S. Colombié. Analysis of Metabolic Networks Using Minimal Cut Set Computing (Abstract & Poster). *Session on Metabolic Pathway Analysis at ISGSB 2012* (Groningen, The Netherlands on September 24, 2012). Available at: http://goo.gl/zOUBTb
- M. Beurton-Aimar, N. Parisey, F. Vallée, **T. V. N. Nguyen**, S. Colombié. Metabolite Hubs to Structure Multi-Pathway Networks (Poster). *71st Harden Conference Metabolic Pathway Analysis 2011*(19–23 September 2011 University of Chester, UK). Available at: http://goo.gl/KVylwU

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE DE BORDEAUX
SPÉCIALITÉ : INFORMATIQUE

Par

Vu Ngoc Tung NGUYEN

Analyse des graphes de reactions biochimiques avec une application au réseau metabolique de la cellule de plante

Soutenue le 3 février 2015
(La version résumé)

Membres du jury :

Mme. Anne SIEGEL	DR	Université Rennes 1	Rapporteur
M. Jérémie BOURDON	HDR	Université de Nantes	Rapporteur
Mme. Marie BEURTON-AIMAR	HDR	Université de Bordeaux	Directeur de thèse
M. Pascal DESBARATS	Professeur	Université de Bordeaux	Examinateur
M. Fabien JOURDAN	Docteur	UMR 1331 INRA Toxalim	Examinateur
M. Dominique ROLIN	Professeur	UMR 1332 INRA BP 81	Président du jury
Mme. Sophie COLOMBIÉ	Docteur	UMR 1332 INRA BP 81	Co-encadrante

Introduction

Un réseau métabolique est constitué d'un ensemble de réactions (équations) qui décrivent une suite de transformations biochimiques. Jusque très récemment, l'échelle des réseaux étudiés se situait au niveau d'une voie métabolique. Bien que certaines voies puissent être relativement complexes, de l'ordre d'une dizaine de réactions impliquées, le raisonnement conduit pour leur analyse, se basait sur des algorithmes supposant un comportement `"linéaire"`, c'est à dire que les cycles étaient éliminés et que lorsque deux voies, deux branches, étaient possibles, chacune était analysée séparément. Dès que les biologistes ont désiré réaliser ces analyses à l'échelle d'un organisme (ou d'un organelle) il est devenu indispensable de repenser les méthodes et plus encore les outils pour conduire ces analyses. En effet ce changement d'échelle provoque un changement drastique du niveau de complexité du réseau étudié et pas seulement un accroissement quantitatif du nombre de réactions à analyser. Un réseau, quel qu'en soit sa nature - réseau social, routier, grille de processeur, processus industriels, etc, peut-être modélisé par un graphe, orienté ou non. Les outils mathématiques ou informatiques dédiés aux graphes sont donc utilisables pour modéliser et analyser les réseaux biologiques.

Dans cette thèse, nous décrirons dans un premier temps les spécificités des réseaux métaboliques et le type de graphe adéquat à leur modélisation. Puis nous étudierons les différentes formalisations des graphes d'interactions et nous montrerons que la méthode des modes élémentaires de flux est un outil puissant pour analyser ces graphes à l'échelle des systèmes. Nous aborderons également les ensembles de coupes minimales, outils complémentaires aux modes élémentaires de flux. La dernière partie de cette thèse sera consacrée à une extension de cette méthode que nous proposons. Cette extension nous permet de définir des modes élémentaires de métabolites. Toutes les méthodes ont été utilisées sur plusieurs réseaux métaboliques, 3 réseaux qui modélisent le métabolisme mitonchondrial dans différents tissus : muscle, foie et levure, et un réseau qui modélise le métabolisme central carboné des plantes. Pour cet exemple, nous déclinerons plusieurs situations suivant les différentes productions de sucre ou d'acides aminées qui ont été étudiées.

Description du graphe d'interactions

Traditionnellement, l'analyse d'un réseau métabolique consiste à réunir un ensemble de réactions de la forme :

$$Reaction\_i : substrat1 + substrat2 = produit1 + produit2$$

Cette réaction décrit la transformation biochimique des deux métabolites *substrat*1 et *substrat*2 en deux autres métabolites *produit*1 et *produit*2. On peut associer un nom à cette réaction, la description du réseau sera donc une liste de réactions similaires à celle ci-dessous.

Nom Réaction	Substrats		Produits
Glucokinase :	Glucose + ATP	=	Glucose-6P + ADP
Isomerase :	Glucose-6P	=	Fructose-6P
Fructokinase :	Fructose-6P + ATP	=	Fructose-6biPhosphate + ADP

Puisque l'ensemble des réactions à l'échelle d'un organisme peut être très grand, on décompose cet ensemble en unité fonctionnelle appelée *voie métabolique*. Cette décomposition, parfois arbitraire, fait appel au concept de fonction biologique. Pour simplifier, on peut définir une fonction biologique comme un ensemble ordonné de réactions concourant à un même objectif. Par exemple la production de sucre (glucose) pour la *glycolyse*.

Réseau et graphe : L'outil naturel en informatique pour représenter des interactions entre différents éléments est le graphe. Un graphe est défini par un ensemble fini de sommets ou noeuds V (ou vertices) et un ensemble E d'arêtes (ou edges) avec $E \subseteq VxV$. Les arêtes représentent les

relations entre les sommets ; les arêtes et les sommets peuvent être étiquetés. Les arêtes peuvent également être valuées, on parlera alors de poids. Un graphe peut être orienté ou non et supporter plusieurs types de sommets. La question de représenter un réseau biologique par un graphe pose la question du choix des entités biologiques qui seront associées aux sommets et aux arêtes. Dans le cadre du métabolisme, il existe plusieurs possibilités. Les sommets peuvent être les réactions, on parlera alors de graphes de réactions, ou bien les métabolites, nommé dans ce cas graphes de métabolites [1], c'est la représentation classique que l'on peut trouver dans la littérature en biologie. On peut aussi créer un graphe appelé bi-partie avec deux types de sommets, les métabolites et les réactions. Lorsque les sommets représentent uniquement des métabolites, les réactions sont positionées sur les arêtes, c'est la représentation choisie dans la figure 1. Comme on peut le voir dans cette figure, dès que la réaction a plus d'un substrat et un produit, une situation très fréquente, le graphe généré est appelé *hypergraphe*. Si cette structure est aisément compréhensible visuellement, son traitement par des méthodes algorithmiques de la théorie des graphes est plus complexe, aussi on traduira le plus souvent un *hypergraphe* par un graphe bi-partie, explicitant l'association de plusieurs substrats dans une réaction ou la génération de plusieurs produits.

C'est le choix qui a été fait par les différents projets internationaux de représentation de connaissances sur les réseaux métaboliques comme KEGG (Kyoto Encyclopedia of Genes and Genomes) ou MetaCyc (Encyclopedia of Metabolic Pathway). La figure 1 montre à nouveau la chaîne de la glycolyse telle qu'elle apparaît sur le site de KEGG, les réactions sont les noeuds rectangulaires, les noms des réactions sont insérés dans ces rectangles, les métabolites sont symbolisés par les petits noeuds ronds, leur nom est inscrit à coté de ce rond. Les flêches sur les arêtes permettent de spécifier la réversibilité des réactions, information importante pour comprendre le jeu de contraintes qui s'exercent sur les interactions.

Graphes bi-partie : Un réseau de Petri [2] est un modèle bien connu en informatique de graphe bi-partie qui permet la simulation du fonctionnement d'un réseau sur un modèle de production/consommation. Plusieurs auteurs [3, 4] ont montré l'intérêt de cet outil pour la modélisation des réseaux métaboliques car un élément important de la définition de ces réseaux est qu'ils décrivent la consommation de molécules (les substrats) et la production de nouvelles molécules (les produits) qui deviendront à leur tour les substrats d'autres réactions. Les réseaux de Petri sont donc particulièrement adaptés pour représenter ces phénomènes surtout lorsqu'on désire simuler le fonctionnement d'une ou plusieurs voies métaboliques interagissant et mises en concurrence pour l'utilisation de molécules communes. Malgré ces avantages, ce n'est pas l'outil que nous avons retenu pour nos études car ainsi que nous l'avons dit, les réseaux de Petri sont utilisés en simulation et notre travail sur l'analyse des réseaux métaboliques concernent plutôt les aspects statiques : structure, topologie pour lesquels les réseaux de Petri ne sont pas obligatoirement les plus adaptés. Toutefois, nous verrons qu'il existe des liens forts entre les outils que nous avons utilisés, les modes élémentaires de flux, et certaines propriétés des réseaux de Petri.

Complexité

Un des éléments fondamentaux de la complexité d'un réseau biologique est la concurrence à laquelle se livrent différentes réactions pour consommer le même métabolite mais aussi le fait que le même métabolite peut être produit par différentes réactions. Une première approche de la mesure de cette complexité peut être obtenue par différents éléments de cardinalité des noeuds comme le nombre de substrats/produits participant à une réaction donnée ou bien, le nombre de réactions différentes reliées au même métabolite. Si l'on considère un réseau métabolique comme une graphe bi-partie, c.-à-d. ayant deux types de nœuds, l'arité moyenne suivant les types est un bon indicateur de la différence de complexité percue intuitivement, suivant qu'on considère le réseau des réactions ou des métabolites. Bien qu'il n'existe pas de règle sur le nombre de métabolites impliquées, substrats ou produits, par réaction, l'expérience montre que le plus souvent l'ordre de grandeur du nombre de molécules impliquées se situe entre 2 et 5/6. L'arité moyenne des nœuds réactions varie donc

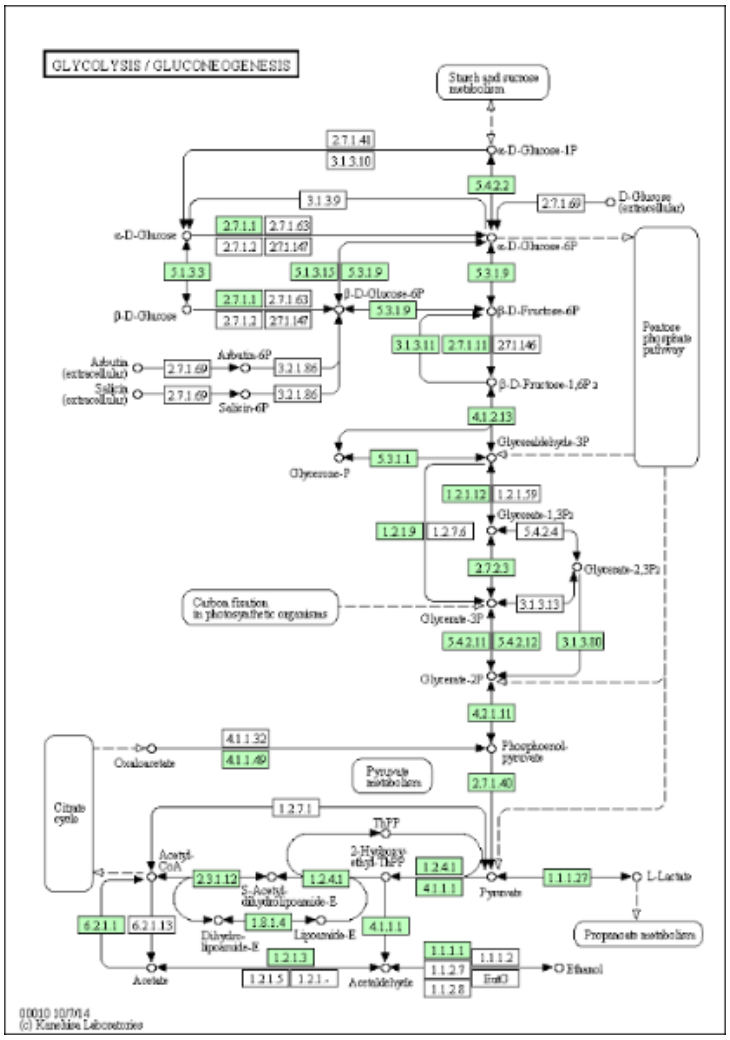


FIGURE 1 – Chaîne de la glycolyse dans la base de donnée KEGG

peu et dans nos exemples de réseaux, on peut constater que l'arité moyenne des nœuds réactions est indépendante de la taille du réseau. Il en est tout autre pour l'arité des nœuds métabolites qui peut se révéler drastiquement différente de celle des nœuds réactions. Ces métabolites fortement utilisés dans le réseau sont généralement appelés métabolites *"hubs"* en ceci qu'ils deviennent des incontournables au moment de calculer le comportement du système.

Les modes élémentaires de flux

Les premiers travaux de notre équipe sur l'utilisation des modes élémentaires de flux (*efms*) dans le cadre de l'étude du métabolisme énergétique de la mitochondrie ont fait l'objet de la thèse de Sabine Pérès. Actuellement, nous nous focalisons sur l'étude du métabolisme carboné de la plante.

La méthode d'identification des modes élémentaires de flux d'un réseau métabolique consiste à déterminer les voies métaboliques admissibles de ce réseau à partir de sa matrice de stochiométrie. Les seules informations utilisées par cette méthode sont la topologie du réseau (coefficient de stochiométrie, réversibilité/irréversibilité des réactions) et ne nécessite pas de connaissance des paramètres cinétiques des réactions. On retiendra comme principe de base de cette méthode qu'elle détermine **les chemins uniques et minimaux** du graphe en respectant la contrainte que le réseau métabolique doit être à l'état stationnaire. Cette analyse topologique permet de caractériser des propriétés du réseau comme la robustesse du réseau (ou son niveau de redondance) [5], les réactions qui opèrent toujours (ou jamais) ensemble...La recherche de voies métaboliques ou suites de réactions correspondant à une fonction biologique a longtemps été considéré comme triviale dans la mesure ou les voies considérées correspondaient aux ensembles de réactions (le plus souvent de l'ordre d'une dizaine de réactions) bien connus dans la littérature. Le passage à l'échelle du système oblige à considérer désormais des ensembles pouvant aller jusqu'à plusieurs centaines de réactions. Ceci conduit inéluctablement à la production de plusieurs milliers de solutions. Stelling et al. [5] ou Wilhelm et al. [6] ont étudié les conséquences de tels résultats en terme de mesure de robustesse des réseaux et apporté un nouvel éclairage sur la façon de considérer la robustesse des fonctions biologiques.

Le tableau 1 ci-dessous résume pour chacun des 4 réseaux que nous avons étudiés le nombre de réactions et de métabolites qui les composent et le nombre d'*efms* que nous avons trouvés.

TABLE 1 – Nombre de réactions, métabolites (total et internes) pour les réseaux de la mitochondrie : muscle, foie, levure, et pour le réseau du métabolisme central de la plante.

Noms	Nb. Réactions	Nb. Tot Métabo.	Nb. Métabo Int.	Nb. EFMS
Mito. Muscle	37	52	31	3 253
Mito. Foie	44	61	36	2 307
Mito. Levure	40	59	34	4 637
Plante	78	70	55	114 614

Les calculs des *efms* ont été obtenus grâce au logiciel regEfmtool [1]. Cette nouvelle version du logiciel Efmtool [2] [7] permet de calculer très efficacement de très grand réseau, éventuellement en utilisant des règles logiques de contraintes. Si historiquement ces calculs étaient réalisés avec l'aide du logiciel metatool puis de sa nouvelle version CellNetAnalyser, les limitations dûes à l'implémentation MatLab des algorithmes rendent ce logiciel très peu utilisable pour les réseaux de grandes tailles. Jungreuthmayer et al [8] ont montré l'intérêt de l'implémentation de regEfmtools

1. téléchargeable à partir de la page `http://www.biotec.boku.ac.at/regulatoryelementaryfluxmode.html`
2. téléchargeable à partir de la page `http://www.csb.ethz.ch/tools/efmtool/`

TABLE 2 – Nombre de réactions, métabolites (total et internes) pour les réseaux de la mitochondrie : muscle, foie, levure, et pour le réseau du métabolisme central de la plante.

Noms	Nb. EFMs	Long. Moyenne	Long. Min	Long Max
Mito. Muscle	3 253	17	2	23
Mito. Foie	2 307	16	2	24
Mito. Levure	4 637		4	22
Plante	114 614	37	2	53

dont les temps de calcul sont de l'ordre de quelques dizaines de minutes quand l'implémentation MatLab requière plusieurs heures, quand les calculs se terminent, ce qui n'est pas toujours le cas.

Malgré tous les problèmes causés par la génération de ce grand nombre d'*efms*, nous tenons à souligner leur réel intérêt en rappelant que dans la thèse de Sabine Pérès [9], il a été montré que dans l'ensemble des efms des 3 réseaux modélisant le métabolisme mitochondrial, il existe plusieurs *efms* correspondant au mutant décrit par Swimmer et al. [10]. Ce mutant permet de produire de l'`ATP` grâce au cycle de Krebs (réaction `R12`) en l'absence d'`ATP synthase` (réaction `R3`). Trouver des *efms* correspondant à des voies *alternatives* prouve formellement que ces voies sont valides dans le réseau et donc peut conforter les résultats biologiques en éloignant le spectre du résultat obtenu par hasard ou erreur de mesure.

Traitement des résultats obtenus

Le calcul des modes élémentaires de flux d'un réseau métabolique donné fournit une nouvelle vision de ce graphe en permettant par exemple d'expliciter les *"shunts"* ou les solutions alternatives existants. De nombreux travaux tentent actuellement de rendre l'analyse plus aisée en découpant par exemple le réseau en modules plus petits [11]. Si cette solution rend parfois les résultats plus intelligibles, elle a l'inconvénient de ne pas être complète puisque bien évidemment les solutions inter-modules (qui ne sont pas obligatoirement la somme des solutions de chaque module) ne sont pas données. Il apparaît donc que la mise en oeuvre d'outils d'analyse automatique des ensembles d'efms obtenus est indispensable pour être réellement utilisable dans le cas des réseaux faisant intervenir plusieurs voies.

Analyse statistique

L'analyse de grandes masses de données est très généralement réalisée au moyen de statistiques descriptives qui permettent de mieux appréhender les résultats obtenus. Dans cette optique, nous avons réalisé pour chaque réseau métabolique étudié, un ensemble de traitement afin de caractériser les résultats obtenus lors du calcul des *efms*.

Calcul des longueurs moyennes Les *efms* étant des chemins minimaux, leur longueur est un bon indicateur de la somme des transformations nécessaires et suffisantes pour aller d'un métabolite *entrant* à un métabolite *sortant* car il n'y a pas à craindre de *bruit* causé par des redondances ou cycles. Nous pouvons observer non seulement une certaine variété entre les 3 exemples mitochondriaux mais surtout lorsqu'on analyse les résultats obtenus pour le réseau de la plante, que la longueur évolue avec la taille du réseau. Ce résultat n'est pas forcément évident car augmenter le réseau signifie en général ajouter des voies métaboliques, encore une fois souvent étudiées séparément, et non étendre chacune de ces voies. On peut expliquer cette augmentation de la taille des *efms* par le fait que l'on doit équilibrer les métabolites, y compris ceux souvent négligés comme le `CO2` ou l'`ATP`, et qu'en ajoutant des réactions on ajoute très souvent de nouvelles contraintes sur ces métabolites.

Calcul des occurences des réactions Pour mieux caractériser la structure d'un réseau, on peut examiner le taux de participation d'une réaction à l'ensemble de solutions obtenues par le calcul des efms. On peut alors s'intéresser aux réactions toujours (ou massivement) présentes qui pourraient être assimilées à des sortes de *hubs* dont l'activité serait des points de contrôle du réseau. Les réactions ne participant à aucun efm sont également intéressantes puisque cela signifie qu'aucun chemin valide dans le graphe ne peut les utiliser. Cela pose alors la question de la validité de la description du réseau. A cette occasion, nous soulignons que la mise au point de cette description : choix des métabolites internes ou externes, choix de la réversibilité ou non des réactions, est un point essentiel de la modélisation des réseaux métaboliques et que le calcul des efms est un outil extrêmement utile pour vérifier/valider cette modélisation. En effet, en détectant ainsi des réactions ne pouvant jamais participer à un chemin équilibré, ce calcul permet d'identifier des connexions dans le graphe qui ne sont pas valides. Il n'est pas possible d'envisager de découvrir ces problèmes simplement en *"regardant"* le réseau car le graphe est d'une taille trop importante pour cela.

Analyse des équations bilan. Il est possible d'obtenir à partir d'un *efm*, l'équation bilan qui lui correspond. Le terme équation bilan doit ici être pris au sens biochimique, c'est l'ensemble des métabolites externes en entrée, nécessaires à la réalisation de l'efm et l'ensemble de ceux qui sont produits. Nous avons analysé cette information car il est intéressant de noter que bien que chaque *efm* soit unique, cela conduit à des doublons dans l'ensemble des équations bilan [3] apportant ainsi une preuve irréfutable que des ensembles différents de réactions (formant des voies valides différentes) conduisent bien à des ensembles de métabolites d'entrée/sortie identiques. Ainsi, dans le cas de mesure de flux métaboliques, il est indispensable de prendre en compte que la seule mesure des métabolites externes se garantit pas l'identification des protéines qui ont été activées. C'est aussi la preuve que lorsque certaines protéines sont *non disponibles* pour effectuer une réaction, que ce soit pour des problèmes de conformation ou parce que l'ensemble des substrats nécessaires ne sont pas accessibles, il est tout à fait possible qu'une *"variation"* de la voie métabolique se mette en place de façon plus ou moins permanente. Pour les réseaux étudiés, en moyenne 4 à 5 efms exhibent la même équation bilan, avec bien sûr des efms qui restent uniques et un maximum du nombre d'efms ayant la même équation bilan pouvant aller jusqu'à 10. C'est cette observation qui nous a conduit à considérer les efms au travers des métabolites qu'ils utilisent.

Ensembles de réactions communs à différents efms Le calcul des efms permet d'identifier des groupes de réactions qui sont toujours associés dans un chemin valide (appelés *subsets* dans le logiciel metatool). Bien qu'en général limité à un petit nombre de réactions, cela permet tout de même d'obtenir quelques simplifications du réseau. Dans nos réseaux, nous avons trouvé pour le muscle, le foie, la levure et la plante, resp. 7, 8, 6, 12 *subsets* réduisant le nombre de réactions à resp. 26, 28, 26, 52. Si des réactions ne sont pas toujours associées dans un efm, elles peuvent l'être souvent, construisant ainsi des *motifs* de réactions communs à un groupe d'efms. L'identification de ces motifs fait l'objet de la section suivante.

Recherche des motifs dans les efms

Il existe un grand nombre de méthodes de classification qui permettent de construire des ensembles en fonction de critères de similitude. Des méthodes tel que le clustering hiérarchique sont couramment utilisées dans des domaines variés - on citera la génomique ou la phylogénie dans le domaine de la biologie.

Malheureusement, les caractéristiques même des modes élémentaires de flux : uniques et minimaux, en font des éléments difficiles à classer par les méthodes classiques. Par exemple si l'on

3. On notera que le logiciel Metatool a choisi de ne pas citer les métabolites qui sont à la fois en entrée et en sortie comme cela est généralement la norme en biochimie. Cette remarque est importante car deux bilans peuvent sembler identiques alors que ces métabolites équilibrés en entrée/sortie ne sont pas les mêmes. Il faut donc être vigilant sur ce point.

considère les méthodes de clustering classiques qui s'appuient généralement sur la construction d'ensembles disjoints, tenter de réaliser ce type de construction avec des *efms* se révèle quasiment impossible et le plus souvent fournit suivant notre expérience, un résultat de peu d'intérêt. En effet si l'on considère dans le graphe d'interactions, d'une part leur propriété d'être uniques et minimaux et d'autre part le fait que le nombre de solutions soit très grand relativement au nombre d'éléments, il est évident qu'un certain sous-ensemble de réactions est commun à différents *efms*. Un rapide test sur d'autres outils classiques comme la construction de treillis de gallois, se révèlent tout autant décevant, car l'explosion combinatoire du nombre de sous-ensembles interdit de tel calcul sur les ensembles d'*efms* de la taille de ceux que nous manipulons.

Toutefois désirant obtenir une classification des nos efms, nous avons conservé l'*idée* de trouver une méthode de type clustering qui soit utilisable. Utiliser de telles méthodes suppose la définition d'une métrique comme critère de ressemblance entre deux éléments. Le codage de la présence ou de l'absence d'une réaction dans un efm est codée par une valeur 0 ou 1 mais comme la réaction peut être utilisée de façon réversible dans l'efm, la valeur -1 est utilisée pour coder cette situation. Nous désirons un critère qui prenne en compte ce cas et aussi le fait que deux efms de longueur 3 ayant 2 réactions en commun, sont plus ressemblant que deux efms de longueur 2 ayant 1 une réaction en commun.

Nouvelle approche basée sur les coupes de graphes

Des travaux récents on ouvert une nouvelle voie dans l'analyse des voies métaboliques grâce à un calcul dual des modes élémentaires : le calcul des coupes minimales du graphe d'interactions. Cette thèse, porte en partie sur l'étude de cet outil.

Le calcul de *"Minimal Cut Sets"* ou MCSs, intègre la même hypothèse que les modes élémentaires de flux en ce qui concerne l'état stable du réseaux, mais au lieu de calculer les chemins possibles, il s'agit alors de calculer les ensembles minimaux de réactions qui déconnectent ce graphe. Il est possible de demander ce calcul pour une fonction objective ou sur l'ensemble du graphe.Le pari est que cet ensemble sera plus petit que celui des modes élémentaires, mais aussi que la taille des MCSs sera en moyenne plus petite que celle de EFMs et donc permettra une analyse plus aisée.

TABLE 3 – **Comparison of the number and the length of EFMs and MCSs**.

Network	**Nb. EFMs**	**Nb. MCSs**	**Nb. MCSs with** `Glc_up`
Vss	22,469	13,901	15
Vac_f	34,752	14,446	15
Vac_g	1,246	562	561
Vac_s	19,392	14,473	15
Vgl_out	19,608	5,500	87

Nous avons réalisé le calcul des MCSs sur nos différents réseaux. La table 3 montre que pour des réseaux dont le nombre de EFMs n'est pas gigantesque, de l'ordre de quelques milliers, nous n'observons malheureusement pas de diminution du nombre d'éléments à observer. Toutefois, dans le cas du réseau de la plante dont le nombre de EFMs dépasse la centaine de milliers, non seulement le nombre de MCSs est inférieur mais surtout la taille des MCSs ne semble pas croître avec la taille du réseau, ce qui nous semble être le résultat le plus intéressant de cette méthode. Malheureusement la recherche de motifs communs grâce à l'algorithme ACOM ne donne pas de résultat satisfaisant, ceci est très probablement dû à la petite taille des MCSs ne permettant pas

la même liberté sur les paramètres de cet algorithme et rendant son réglage très délicat. Nous avons réalisé des statistiques descriptives des MCSs obtenus. Ainsi il est toujours intéressant de répertorier les réactions qui n'appartiennent jamais à un MCS. Cela signifie que le réseau ne peut jamais être déconnecté au moyen de cette réaction. On peut donc en déduire que construire un mutant qui inhibirait ces réactions n'aurait pas d'effet sur le comportement général du réseau métabolique. Les réactions toujours présentes dans les MCSs sont par ailleurs indispensables au fonctionnement du réseau, mais ceci peut bien sûr être également observé dans les efms. Les couples ou les triplets de réactions (on ne considèrera pas les MCSs de taille 1 dont l'interprétation est triviale) sont intéressants à étudier car ils fournissent un résultat très facile à exploiter pour les biologistes. Un couple ou un triplet de réactions qui constituent un MCSs peut couper toutes les voies possibles dans un réseau, cette information permet de mieux comprendre l'activité de ce réseau surtout si ces réactions ne sont pas directement reliés aux mêmes métabolites. Pour mieux expliquer ceci voici un exemple très simple du TCA cycle (ou cycle de Krebs).

Étude de cas : production de sucres et acides aminées dans le fruit de tomate

A partir des résultats obtenus à la fois dans le calcul des EFMs et des MCSs sur le réseau donné en annexe, nous avons sélectionné les EFMs permettant la production de 6 différents substrats ayant un intérêt dans l'étude du métabolisme du fruit de tomate dans le cas où il n'y a pas d'entrée de `Glucose` (réaction Glc_up). Pour ce faire, nous avons sélectionné pour chaque cas, les EFMs contenant la réaction responsable de cette production. Ces substrats sont Glucose, Fructose, Sucrose, Glutamine, Starch et les réactions concernées sont respectivement : Vac_c, Vac_s, Vac_m, Vss, Vgl_out.

La table 3 montre pour chaque cas les effectifs de EFMs concernés. En ce qui concerne les MCSs, nous avons sélectionné les MCSs qui contiennent Glc_up (puisque celui-ci est bloqué) et la réaction ciblée. A partir du résultat des MCSs de taille 2, nous avons identifié 8 réactions qui participent toujours à la production des 5 metabolites d'intérêt en absence d'entrée de glucose. Ces réactions peuvent être considérées comme le coeur du réseau. Ces réactions sont : **Vpgi**, **Vfbp**, **Vpgi_p**, **Vrbco**, **Tg6p**, **Vald**, **Vriso_p** et **Vepi_p**. En analysent les MCSs de taille 3, ajoutent une liste de 5 réactions qui sont ensuite une des alternatives possible pour les différent chemins possibles. L'utilisation conjointes des EFMs et des MCSs nous permet donc d'identifier des réactions hubs dans ce réseau.

Pour terminer ce chapitre, nous voudrions souligner l'importance de la qualité du code des différents outils utilisés. Les versions les plus récentes des concepteurs de la méthode des modes élémentaires ont fait le choix de privilégier des versions utilisants un environnement Matlab, malheureusement peu adéquate pour supporter les calculs lourds. Non seulement cette bibliothèque n'est pas très rapide mais surtout malgré une documentation affirmant que dans sa version unix, la taille de la mémoire n´était limitée que par la mémoire disponible sur la machine, nous avons constaté qu'il n'en était rien. Les calculs sur le réseau de la plante sont quasiment impossible à obtenir avec les versions de CellNetAnalyser sous Matlab. Fort heureusement, il existe d'autres versions du calcul des EFMs, entre autre celle écrite en langage java par Marco Terzer [7] mais elle est peu documentée. Plus récemment, Christian Jungermeyer [8] a produit une bibliothèque de fonctions intégrant EFMtools et une extension qui permet d'écrire un ensemble de règles logiques pour calculer les EFMs avec des contraintes fonctionnelles. Dans le même environnement, mais cette fois écrit en langage C, on dispose aussi du calcul des MCSs et ce de façon très performantes. L'ensemble des calculs regEFMtools et mcsCalculator, font en général passer les calculs de plusieurs heures avec CellNetAnalyzer (quand ils terminent) à moins d'une minute.

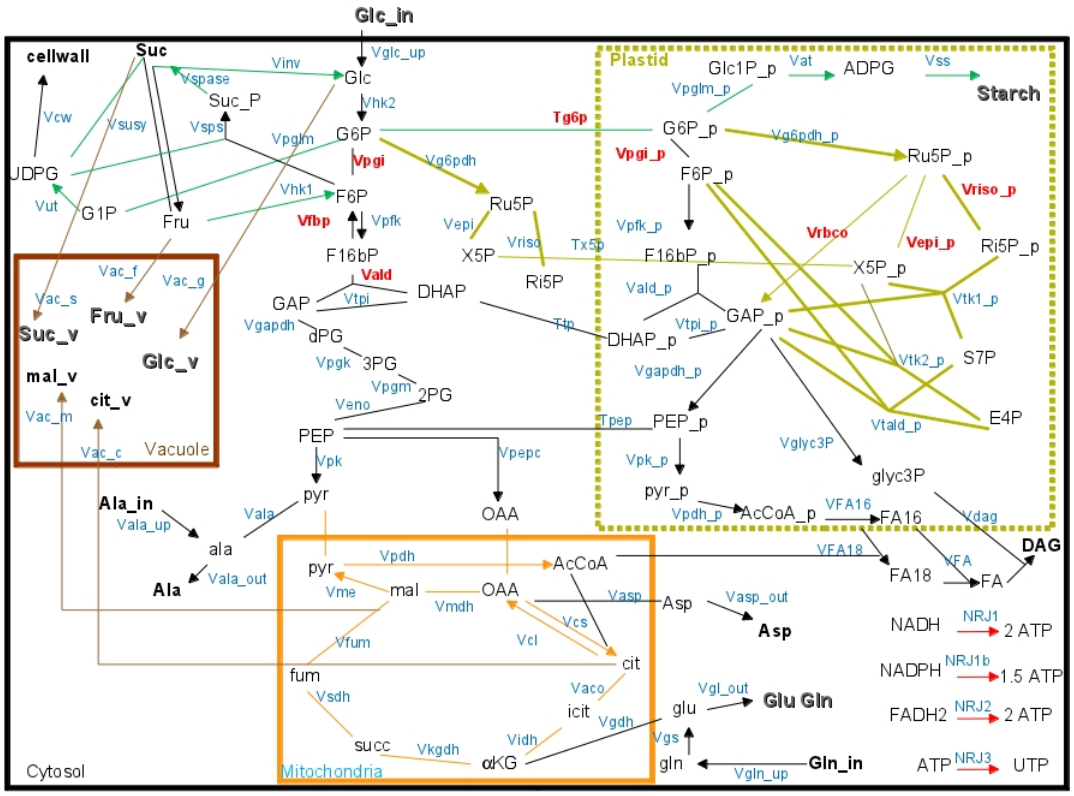


FIGURE 2 – **Enlarged metabolic network of a heterotrophic plant cells with 8 mandatory reactions highlighted.**

Conclusion

L'analyse de la structure statique des réseaux permet de mieux identifier le niveau de complexité auquel se situe les réseaux métaboliques. En effet, le passage de l'étude d'une réaction à celle de la voie métabolique puis d'un ensemble de voies constituant un métabolisme ne génère pas une complexité qui croit linéairement mais bien exponentiellement bien que l'ajout de noeuds modifie peu les paramètres classiquement étudiés en théorie des graphes comme l'arité moyenne des noeuds ou le diamètre du graphe. Les outils comme la recherche de chemins minimaux dans le graphe, les EFMs, permettent d'identifier cette complexité mais les résultats obtenus restent encore difficile à analyser entre autres à cause de leur taille. La combinaison de l'analyse des EFMs et des MCSs permet d'identifier les réactions les plus essentiels pour produire un metabolite d'intérêt. Notre analyse du réseau du fruit de tomate a montré que malgré la taille des données à manipuler il était possible d'en extraire des informations utiles qui peuvent ensuite être prise en compte dans l'interprétation des expériences biologiques qui sont conduites.

Références

[1] A. Wagner and D. A. Fell, "The small world inside large metabolic networks," in *Proceedings of the Conference of The Royal Society in London B*, vol. 268, Apr. 2001, pp. 1803–1810.

[2] J. L. Peterson, "Petri Nets," *ACM Comput. Surv.*, vol. 9, no. 3, pp. 223–252, Sep. 1977. [Online]. Available : http://doi.acm.org/10.1145/356698.356702

[3] I. Koch and M. Heiner, "Petri Nets," in *Analysis of Biological Networks*. Wiley, 2008, ch. 7, pp. 139–179.

[4] E. Grafahrend-Belau, F. Schreiber, M. Heiner, A. Sackmann, B. H. Junker, S. Grunwald, A. Speer, K. Winder, and I. Koch, "Modularization of biochemical networks based on classification of Petri net t-invariants." *BMC Bioinformatics*, vol. 9, no. 1, p. 90, Jan. 2008. [Online]. Available : http://www.biomedcentral.com/1471-2105/9/90

[5] J. Stelling, U. Sauer, Z. Szallasi, F. J. Doyle 3rd, and J. Doyle, "Robustness of cellular functions." *Cell*, vol. 118, no. 6, pp. 675–685, 2004.

[6] T. Wilhelm, J. Behre, and S. Schuster, "Analysis of structural robustness of metabolic networks," *Systems biology*, vol. 1, no. 1, pp. 114–120, 2004.

[7] M. Terzer and J. Stelling, "Large-scale computation of elementary flux modes with bit pattern trees," *Bioinformatics*, vol. 24, no. 19, pp. 2229–2235, 2008. [Online]. Available : http://bioinformatics.oxfordjournals.org/content/24/19/2229.abstract

[8] C. Jungreuthmayer, D. E. Ruckerbauer, and J. Zanghellini, "regEfmtool : Speeding up elementary flux mode calculation using transcriptional regulatory rules in the form of three-state logic," *Biosystems*, vol. 113, no. 1, pp. 37–39, 2013. [Online]. Available : http://www.sciencedirect.com/science/article/pii/S0303264713000890

[9] S. Pérès, "Analyse de la structure des réseaux métaboliques : application au métabolisme énergétique mitochondrial," Ph.D. dissertation, Université de Bordeaux 2, 2005.

[10] C. Schwimmer, L. Lefebvre-Legendre, M. Rak, A. Devin, P. P. Slonimski, J. P. Di Rago, and M. Rigoulet, "Increasing mitochondrial substrate-level phosphorylation can rescue respiratory growth of an ATP synthase-deficient yeast," *Journal of Biological Chemistry*, vol. 280, no. 35, pp. 30 751–30 759, 2005.

[11] J. Gagneur and S. Klamt, "Computation of elementary modes : a unifying framework and the new binary approach." *BMC bioinformatics*, vol. 5, p. 175, 2004.