Metabolism and Bioinformatics: The Relationship between Metabolism and Genome Structure

BHUSHAN BONDE

A thesis
submitted in partial fulfilment of
the requirements of Oxford Brookes University
for the award of the degree Doctor of Philosophy

Oxford, UK
July 2006

Abstract

The availability of data on genome, proteome, transcriptome and biochemical/metabolic interactions allows integrated study of biological interactions using a systems level approach. One instance of this, is the study of *in silico* metabolic networks. For kinetic studies of metabolism, knowledge of a large number of parameters (rate constants, concentrations etc) is required, and these may not be known exactly or measured experimentally for many biological systems. For such systems, building a structural model of a metabolic system is much simpler as the structural model only needs information about reaction stoichiometry, reaction reversibility and information about the external metabolites of the system. With structural models it is possible to reach conclusions about some systems properties of metabolism. This thesis investigates the applications of a number of structural modelling techniques to large 'omics' scale metabolic networks in order to access their feasibility and usefulness in understanding the physiology of a model organism.

The structural analysis of a metabolic model includes detection of enzyme subsets, dead reactions, orphan and dead-end metabolites, elementary modes and futile cycles. An enzyme subset is a group of enzymes that always operate together in fixed proportion. It was hypothesized that enzymes (reactions) in the same subset should have common genetic regulation of their expression. This hypothesis was tested by comparing the enzyme subsets of models of the *E.coli* metabolic network with the known operons with metabolic function. It was found that few subsets match with known operons in *E.coli*.

The next test was to examine a genome-scale metabolic model of *E. coli* (762 metabolites and 925 metabolic reactions) to study the relationship between enzyme subset and gene coexpression. Microarray data was used to study the coexpression profile of genes. A significant correlation was observed between the enzyme subset and their corresponding gene coexpression. During the work, it was observed that genes involved in coexpression of enzymes in a reaction subset share a similar clustering on genomic level. A significant correlation was observed in enzyme subsets and operons in *E. coli*. In few reaction subsets, non-

adjacent but functionally co-related genes (possible regulons) were identified from microarray gene co-response correlation data.

Another aspect of the present study was to determine the importance of reactions via assessment of the damage to the metabolic network if they are inactivated. A previously reported graph theoretic damage method (Lemke *et al.*, 2004) which uses a graph theoretic approach based on the local connectivity of the network was modified to account reaction reversibility. Its predictions were compared with those derived from two other structural modelling techniques: null space and elementary mode analysis. The null space and elementary mode damage can predict the effect of removal (damage) of a reaction using the global properties of the network rather than using the local neighbourhood connectivity of the system. The elementary mode damage algorithm predicts a more accurate damage set in a small metabolic system, but cannot be applied to large metabolic systems due to the computational complexity. Null space damage prediction gives results closer to elementary damage prediction than the graph-theoretic method, but all three methods give different results.

Acknowledgements

I am massively indebted to Prof. David Fell who gave me the opportunity to pursue my interest in this research area, for sharing his expertise in the field of metabolic modelling and providing inspiration and confidence in my abilities.

I would like to thank Dr. Mark Poolman who enlightened me throughout the course of my PhD and provided me with valuable suggestions and help in writing computer programs and algorithms. I also thank Dr. Tjeerd for his advice at the initial stage of my research work. I thank my colleagues, Dr. Heike Assmus who always helped me to understand the mathematical methods in modelling, my colleagues Harshil Patel and Albert Gevorgyan for their enthusiasm and comments on my work. I am very grateful to Oxford Brookes University for providing financial support and research fellowship for this research.

I would like to express gratitude to Prof. D. Steinhauser of MPI, Golm, Germany for providing the co-response data from the CSB database and Prof. B. Palsson for providing me with the metabolic model data for my study.

I would like to thank Dr. Rekha Singhal, for proofreading and suggestions on this dissertation and to Parag Saudagar, for sharing his data and ideas about the work. To Farida, for assisting me with the administrative tasks necessary for completing my doctoral program and being there to listen to all my concerns.

Many people have aided in the development of this dissertation and I would like to express my appreciation for their dedication, knowledge and assistance. The Oxford clique of Bhaskar, Kiran, Srinivas, Bhumin and Niranjan, with whom I spent many hours of debates from science to politics, I owe them many thanks for their support. Since I have obtained support from numerous people, rather than risking forgetting any body I would like to say that I cherish each and every contribution to my development in both my personal and scientific life.

I would like to thank my parents and younger brother for their unfailing love and encouragement. A special thank to Priya, whose constant support and efforts led to the completion of this dissertation.

Mathematics is biology's next microscope, only better;
Biology is mathematics' next physics, only better.
Joel E. Cohen

Dedication

DEDICATED TO

MY LOVING

PARENTS

AND

MY WIFE

'PRIYA'

Contents

List of Figures

.

List of Tables

Chapter 1
Introduction

After the sequencing of the first complete genome sequence of *Haemophilus influenzae* by Fleischmann *et al.*(1995), there has been an exponential rise in the genome sequencing of various other microbial species, plants and animals. Though the genome sequencing was achieved using high-throughput and computational techniques, it led to a new challenge to interpret the meaning of the sequenced genomic data in the post genomic era. Various mathematical and computational tools to interpret the meaning of the sequence information such as comparative genome analysis (Dandekar & Sauerborn, 2002), cluster analysis (Eisen *et al.*, 1998), sequence alignment (Schuler, 2001) and phylogenetic analysis (Lesk, 2002) were developed to understand the biological significance and meaning of the 'omics' data. Such techniques were often used to understand the functional capacity of the organism.

The flow of information in any living cell from DNA to mRNA to proteins is well documented in text books of biochemistry (Lehninger *et al.*, 1993; Stryer, 1995). This gives an impression of a linear flow of genetic information. The central dogma is that the DNA sequence (gene) determines the protein sequence, which in turn determines the protein structure and function. Though the central dogma of life is true, in reality, it is now known that this central dogma is just a part of the highly complex interactions taking place in the living cell.

The systematic study of metabolism in various living organisms started in the late 19th century, and has since then accumulated a large amount of data on biochemical and metabolic interactions in a cell. Although the understanding of the biochemical interactions in a living cell started much earlier, albeit at a slower rate than genome analysis; the rate of study of genomes of various organisms occurred much more rapidly in the last two decade. Nevertheless, serious efforts are needed to investigate the relationship between the

biochemical and genomic interactions. During the process of evolution, genes have undergone mutation, alteration or deletion (due to damage) from one generation to another. Over a long time, genes have altered significantly even in the same species. The metabolic reactions have, however, been retained throughout with fewer or negligible alterations, maintaining the functional integrity of the cell.

Our motivation for the present study was to use the present available knowledge of metabolic reactions to understand the relationship between the metabolism and the genome of an organism.

1.1 *E.coli* as a model organism

Over the last century, research on a small number of organisms has played a pivotal role in advancing our understanding of numerous biological processes. This is because many aspects of biological processes are assumed to be similar in most or all organisms. Therefore, it is much easier to study many biological aspects in one organism than in several organisms. These much-studied organisms are commonly referred to as 'model' organisms, because each has one or more characteristics that make it suitable for laboratory study. The most popular model organisms have strong advantages for experimental research, such as rapid development with short life cycles, small adult size, ease of availability, and tractability, and become even more popular when many other scientists work on them. A large amount of information can then be derived from these organisms, providing valuable data accumulation for the analysis on its metabolism, development, gene regulation, genetic diseases, and evolutionary processes.

Escherichia coli belong to the family of Enterobacteriaceae and is one of the main species of bacteria that live in the lower intestines of warm-blooded animals, including birds and mammals. The name comes from its discoverer, Theodor Escherich. *E.coli* is a part of the normal intestinal flora and is necessary for the proper digestion of food. Few *E.coli* strains can cause food poisoning or

illnesses that are more serious (e.g. *E.coli* O157:H7 outbreaks *via* contaminated beef that was not cooked thoroughly).

E. coli K-12, which has two commonly studied strains MG1655 and W3110 is well studied in microbiology due to its harmless and ubiquitous nature. They are commonly used as a model organism because of the ability to carry out genetic recombination by conjugation (transferring DNA from one cell to another cell) and by generalised transduction (the ability of certain phages to transduce any gene in the bacterial chromosome) (Lederberg & Tatum, 1946). The entire genome sequence of K-12 strain MG1655 was first completed and annotated by Blattner *et al.* (1997), and later reannotated and updated by Serres *et al.* (2001). Recently, a complete snapshot of the sequence and annotations for *Escherichia coli* K-12 MG1655 has been deposited in GenBank (accession no. U00096.2). The overall updates on the highly accurate, but still incomplete annotation was published by Riley *et al.* (2006). Recently during the write-up of this dissertation, Hayashi *et al.* (2006) published a highly accurate draft of the *E.coli* K-12 strains MG1655 and W3110. The regulons and operons in *E.coli* are also thoroughly studied over the past few years. *E.coli* K-12 became the primary model organism of choice for studying basic biology, molecular genetics and physiology of bacteria all over the world, becoming a 'workhorse' for biotechnology. The great scientist Monod once said, "Tout ce qui est vrai pour le Colibacille est vrai pour l'éléphant" meaning "Once we understand the biology of *E. coli*, we will understand the biology of an elephant" (Monod, 1972).

Most of the metabolic and biochemical reactions of *E.coli* that appeared in literature (Riley, 1993) over the last few decades made it the perfect choice to be used as a model organism for numerous wet lab as well as *in silico* studies.

1.2 Motivation

The main objective of the present study was to analyse an *in silico* genome scale metabolic model of a model organism *Escherichia coli* in order to explore the

relationship between metabolic and genomic data. One of the main aims of the project was to identify the challenges and problems involved in the automated reconstruction of metabolic networks from metabolic and genomic databases. Another major objective was to study the structural (or topological) aspect of metabolic networks that can be easily reconstructed without the need of any enzyme kinetic data from the organism. The enzyme kinetic data is necessary for kinetic metabolic models but structural models can be reconstructed using only reaction stoichiometry and reversibility.

The development of methodologies for reconstruction of accurate (error free) models from the various public databases will be discussed in this work. The use of structural models of organisms for the identification of functionally distinct units based on the global properties of the network (rather than using local/connectivity) and understanding the significance of the functional modules with respect to the genome structure of the organism was the final motivation of the project.

1.3 Metabolic modelling

Due to the advances in the applications of the mathematical and the computational studies of metabolism, the term ‘metabolic modelling’ represents the wide range of techniques shown in Figure 1-1.

The criteria of classification shown in Figure 1-1 is based on the computational methods used, however, it might be possible that modelling techniques use more than one computational technique or alternative approaches. For example, elementary modes can be computed by two distinct algorithms, directly from stoichiometry matrix or from the null space matrix of the system. The shaded circle in the Figure 1-1 shows the techniques used or developed in the present study, which focus on the topological (structural) network analysis and these techniques are discussed in detail in Chapter 2.

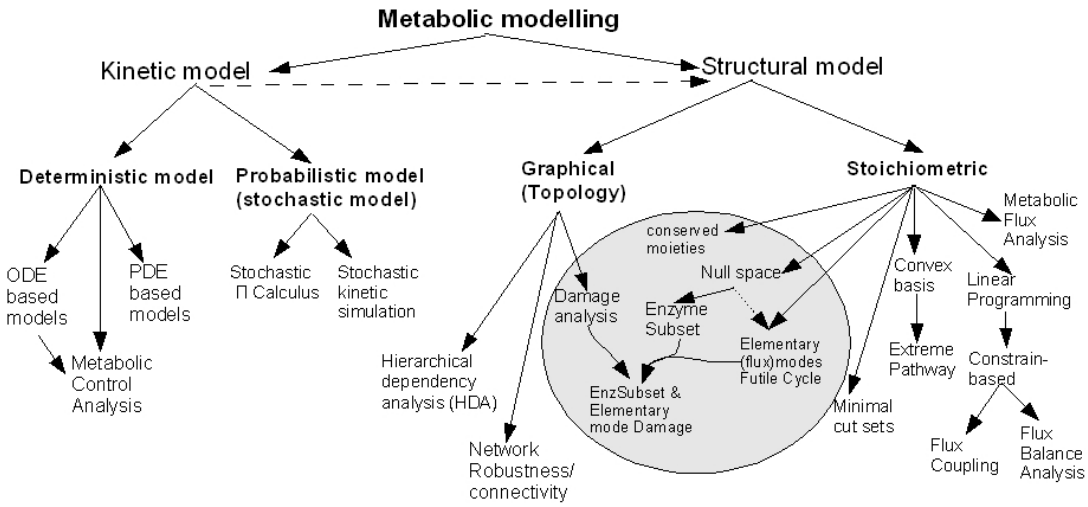


Figure 1-1: General classification of metabolic modelling techniques.
The shaded circle indicates the area of interest for the work discussed in this thesis.

At present, the purpose of building a model is to enable answering one or more of the following open questions from the large structural metabolic model:

- How many routes are possible from one metabolite to another metabolite in a large metabolic system? Which is the most obvious route between the two metabolites? What is the average number of reactions in such routes?

- Does a metabolic network have functionally similar modules or groups of reactions? How can such functionally correlated modules be identified in a complex system?

- Which are the highly connected metabolites (also known as hubs) in the network? What is the significance of such hubs in the metabolic network?

- Which are the important reactions (enzymes) in the metabolic network for a given biological function?

- What is the relationship between the metabolic network and the genome structure of the organism?

Of further interest is the appearance of hierarchical organization of sub networks, i.e. if and when the metabolic system is modular. Would it be possible

to qualitatively identify, and further quantify the size distribution of such independent modules from the networks found in the metabolic system?

The next question is the relevance of modular or substructures of a complex network during the analysis. Should the whole complex network always be taken into account? The idea of modularization of the complex system is not new to engineers, e.g. the task of building an aeroplane involves construction of separate components such as wings and body, and assembling them as a system. Analogically, could metabolic networks be broken into similar functionally distinct parts? How could such functional modules be identified? What criteria could be used to separate the two components from one another? Do functional metabolic modules show some correlation on genome or transcriptome level? Lastly, how to interpret the meaning from such functional modules? Is any correlation possible between functional modules and organisation (or modules) of genome/transcriptome? The quest of such open questions was a major motivation of the present study.

1.4 Outline

Chapter 2 of this thesis introduces the theory of structural modelling. The present method of structural modelling that is based on the stoichiometry matrix is explained in detail.

Chapter 3 describes the reconstruction of genomic scale metabolic models. Identification of problems and challenges in building and modelling such large metabolic networks is discussed in detail. The application of the theory of structural modelling to large genomic-scale metabolic models is also discussed.

Chapter 4 presents the analysis of the substructure of a large genomic scale metabolic model of *E.coli*, its relationship with *E.coli* genome, the relationship between the reaction clusters and gene clusters (operons or regulons).

Chapter 5 explains the relationship between reaction subset (enzyme subset) and the coexpression profile of the genes. It presents the correlation between the reaction clusters with the gene microarray expression data.

Chapter 6 describes the extension of the methodology 'damage analysis' (Lemke *et al.*, 2004) for analysing and studying the metabolic networks. Damage analysis involves the study of a large metabolic network by determining the importance of a reaction by its elimination (i.e. elimination of the reaction or an enzyme responsible for that reaction) from the network, and calculating the score as a number of subsequent reactions damaged due to the knocked or deleted reaction. Depending on the damage score, reactions can be classified as essential (if a reaction causes a deletion of many other reactions leading to cell death) or non-essential. This concept of damage analysis was based on the application of graph theoretic analysis of the metabolic network. This is further extended to two newly developed algorithms based on null space and elementary modes damage where damage by a reaction was analysed using the global properties in the metabolic network. The two algorithms are believed to predict more accurately compared to the graph theoretic damage, and may lead to better understanding of the importance of a reaction in metabolic system.

Chapter 7 includes discussion and possible applications of the techniques developed during the present study. It also includes proposed future work.

Chapter 2
Theoretical background - modelling of metabolic networks

2.1 Introduction

This chapter introduces the various basic techniques used in the modelling of metabolic networks, and the theory related to structural modelling is discussed in detail. For the present work, ScrumPy, a general-purpose metabolic modelling package written in Python (Poolman, 2006a) was used. Therefore, the computer representations of the models used as examples throughout the thesis comply with model definitions of ScrumPy. The theory behind the structural modelling can also be implemented in any other structural modelling software.

2.2 Metabolic models

A metabolic model is a representation of a metabolic system (or network) in a mathematical form. Metabolic models help in capturing and analysing the complex behaviour of the system (or network).

Most of the biological or metabolic reactions take place inside a living cell. Even the simplest free-living, non-parasitic (unicellular) organism contains more than a thousand metabolic reactions. Reconstructing or building a metabolic model involves collection and representation of the metabolic reaction data from a living cell or organelle.

2.2.1 Model definition

Any theoretical model is always based on definitions and rules representing the real data. A metabolic model contains certain chemical entities, such as metabolites, enzymes and transporters.

A. Metabolites

For modelling purposes, metabolites are considered to be of two types - external and internal. External metabolites are the ones that are found outside the frame or boundary of the model. Examples of such external metabolites are:

a. Source metabolites (supplied to the system for its growth or media composition) and sink metabolites (produced or excreted from the model system).

b. Metabolites, such as water, that are always in excess and have concentrations that are virtually independent of the system being modelled, are usually defined as external (Poolman *et al.*, 2004b).

c. Polymeric molecules present in the system are also defined as external, as the exact number of the monomers in such molecules cannot be given in the reaction stoichiometry (e.g. starch or glycogen, DNA, RNA etc).

Metabolites like ATP, NAD, FAD, and NADP could be either internal or external, depending on the objective of the modelling. In the case of large metabolic networks, where the topology of the network is the focus of investigation, one can define the above metabolites as external, thus reducing the connectivity of the complex system without the loss of functionality of the network (Fell and Wagner, 2000). Such metabolites are also called currency metabolites (Papin *et al.*, 2002) and are discussed in further detail in Chapter 6.

All the other metabolites in the model are internal metabolites, and they behave as variables of the model.

B. Reactions

Reactions convert one chemical form of metabolite to another or connect one metabolite to another in the metabolic network. Most (but not all) reactions are mediated in the cell *via* enzymes or transporters, few reactions are spontaneous and do not require any enzyme or mediator. The link between the reaction and enzyme is not always simple. Several enzymes (isoenzymes) may be able to carry out the same reaction in a cell, or a particular single enzyme may have the

ability to carry out various different reactions. The information on the reaction mediator is therefore vital in model building and analysis. Similar additional aspects (e.g. gene protein reaction association) of analysis of the relationship between genome and metabolic structure are discussed in Chapter 4.

In principle, almost all reactions are thermodynamically reversible (Hofmeyr & Cornish-Bowden, 1997), although in practice some reactions can be considered to be irreversible *in vivo*, if they exclusively proceed only in one direction. Under all physiological conditions of substrate and product concentrations, such reactions have a higher rate of forward conversion than of backward conversion. Information of such irreversible reactions is of great importance not only in building the structural model but also in analysing and understanding of the system.

2.2.2 **The steady state system: Concept**

Let us assume a simple system as shown in Figure 2-1, containing only one substance *S* with three reactions. Two reactions, T_1 and T_2 import *S* (source reactions) from outside into the system while T_3 export *S* (sink reaction) to outside of the system. The change in the concentration of *S* depends on the overall (sum of) rates of production (T_1 and T_2) and rate of consumption (T_3) of *S*.

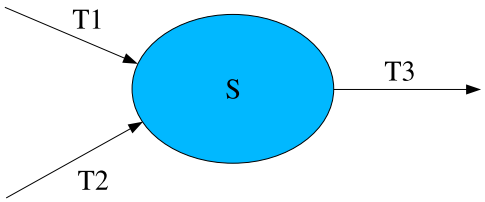


Figure 2-1 A simple system at steady state

At the steady state, i.e. while the system is at equilibrium, the rate of production of substance *S* is always equal to its rate of consumption and the total amount of substance *S* remains unchanged in the system,

$$\frac{dS}{dt} = 0 \qquad (1)$$

The above concept can also be applied to any other system, with more than one substance, and the substances can be interconnected or are inter-convertible (*via* reactions) while a few can be transported across the system boundary.

At the steady state, the concentrations of all the internal metabolites always remain balanced (as the supply to the system is always equal to the demand). If $\overline{S}$ represents the vector of the concentrations of all the internal metabolites, then at the steady state, equation 1 can be written as,

$$\frac{d\overline{S}}{dt} = \overline{0} \qquad (2)$$

where $\overline{0}$ represents a zero (null) vector.

2.2.3 **The stoichiometry matrix**

To understand the concept of the analysis of metabolic networks based on stoichiometric matrix, let us first consider a very simple model system as shown in Figure 2-2A, where external glucose (X_Glucose) is converted to external fructose 1,6 biphosphate (X_F16bP) in 5 reactions. Biologists are familiar with such diagrams (this is a graphical representation of the model system).

The model system can also be written as a list of individual reactions, their reactants and products, as shown in Figure 2-2B (This is a computer representation of the model system for ScrumPy software (Poolman, 2006b)). In this form, a reaction is written as a balanced reaction of reactants and products (the reaction stoichiometry). The stoichiometric coefficients give the molecularity (proportions of number of molecules) of the reactants and products that are involved in the reaction. The stoichiometric coefficients of all the reactions in the system can be considered as vectors and can be arranged in a matrix (see appendix A). The stoichiometry matrix $\boldsymbol{N}$ is a mathematical representation of the system with the dimension $m \times n$ where m (i.e. the number

of rows) is the total number of the internal metabolites, and *n* (i.e. the number of columns) is the total number of reactions of the system[1].

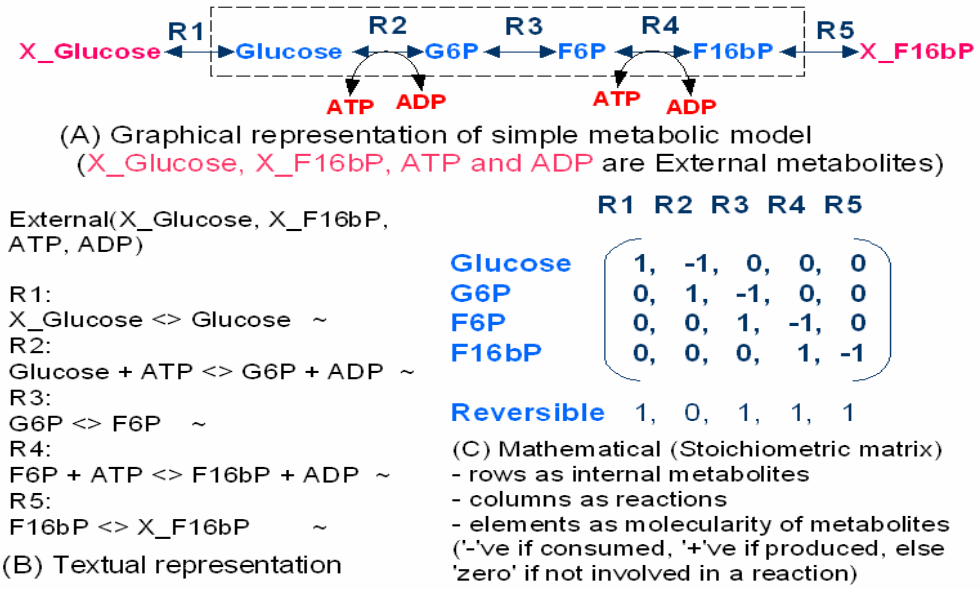


Figure 2-2 Representations of the metabolic model

Each element of the stoichiometry matrix corresponds to the molecularity of the corresponding metabolite in the reaction. A zero element in ***N*** shows that there is no involvement of the corresponding metabolite in the respective reaction. A negative element represents utilisation of the metabolite in the reaction, and a positive element represents production of the metabolite in the reaction. The stoichiometry matrix is constructed with the convention that each reaction is written from 'left to right' (especially for irreversible reactions), and this convention is maintained throughout for all the irreversible reactions in the system as shown in Figure 2-2. Converting a metabolic system into a matrix form has an advantage; it allows various well-known mathematical methods of linear algebra to be carried out on the system under study (see Appendix A).

[1] ***N*** represents the (internal) stoichiometry matrix of the system throughout this dissertation.

2.2.4 **Stoichiometry matrix and steady state concept**

The stoichiometry matrix can be considered as a term in the system of simultaneous linear equations that describes the dynamics of the system and that can be written in the form of three matrices (see Appendix A). If $\bar{v}$ represents the vector whose elements correspond to reaction rates of the metabolic system, then the rate of change of the concentration of each metabolite is given by the fundamental equation as,

$$\frac{d\bar{S}}{dt} = \boldsymbol{N}.\bar{v} \qquad (3)$$

The rate of each reaction (ν_i) depends on the concentrations of the metabolites in the reaction, forming nonlinear kinetic equations of the system. The variables of the system include a) the reaction rates (ν_i) of the system and b) the concentrations of the internal metabolites.

Structural metabolic network analysis is based on the assumption that, on a longer time scale, the concentrations of the (internal) metabolites do not change. This quasi (pseudo) steady state assumption is applied to the fundamental equation. At the steady state, combining equation 2 and equation 3,

$$\frac{d\bar{S}}{dt} = \boldsymbol{N}.\bar{v} = 0 \qquad (4)$$

Equation 4 is the basis of the stoichiometry based structural modelling. The theory of stoichiometric analysis of metabolic models at steady state is well reported in various publications (Clarke, 1981; Cornish-Bowden & Hofmeyr, 2002; Fell & Small, 1986; Heinrich *et al.*, 1977; Heinrich & Schuster, 1996; Hofmeyr, 1986).

2.2.5 **The null space matrix**

To understand and analyse the system, one needs to solve equation 4 of the system (see Appendix A). A simple and trivial solution is $\bar{v} = 0$. However, this solution can not be considered. Though it satisfies the mathematical requirement, it violates the biological properties; all the reactions in the system carry no flux means the overall system is 'dead'. As explained in Appendix A, if

the rank of the stoichiometry matrix (rank(***N***)) is equal to the number of unknown variables (i.e. number of reactions or elements in $\bar{v}$), the system may have a unique solution, else it has an infinite number of solutions. However, even if there are an infinite number of solutions, each equation imposes a constraint that results in identifying relationships between the values of $\bar{v}$. Using linear algebra, it is possible to find the subspace of all possible solutions called the null-space matrix (Hefferon, 2003; Reder, 1988) of the system.

The null space matrix[2] (***K***) (also called as Kernel of ***N***) is the subspace of all vectors satisfying equation 4 (Heinrich & Schuster, 1996). Each column vector k_i of ***K*** represents a possible solution to equation 4 and the set of columns in ***K*** are linearly independent (Reder, 1988). Mathematically ***N*** and ***K*** are related by equation,

$$\boldsymbol{N}\ \boldsymbol{K} = 0 \qquad (5)$$

The null space can be determined either by the Gaussian elimination method (Heinrich & Schuster, 1996) or by Singular Value Decomposition (SVD). The former is simple to implement and easy to understand while the latter is complex and more suitable for very large stoichiometry matrices (Press *et al.*, 1989).

The study of the null space is useful because for any possible steady state of the system, rate of reaction vector ($\bar{v}$) can be expressed as a weighted combination of the vectors in the null space matrix (***K***) which further extends to the concept of elementary modes concept in metabolic modelling .

A model system, as shown in Figure 2-3-A for example, converts glucose into fructose in a cyclic manner. The transport reaction R1 imports glucose while reaction R4 exports fructose from the system. For such a cyclic network, N is given as:

[2] ***K*** will represent the (right) null space of the *N* henceforth in the dissertation.

$$N = \begin{array}{c} \begin{matrix} \text{R1} & \text{R2} & \text{R3} & \text{R4} \end{matrix} \\ \begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & -1 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} \\ \text{Glucose} \\ \text{Fructose} \end{matrix}$$

Using equation 5, the null space of the system can be obtained as:

$$K = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{matrix} \cdots \text{R1} \\ \cdots \text{R2} \\ \cdots \text{R3} \\ \cdots \text{R4} \end{matrix}$$

Though the null space gives all the possible solution spaces of the system, the basis of the null space is not unique, i.e. the set of spanning vectors can be represented in different forms of the matrices. The system in Figure 2-3 can have the following two possible null space bases.

$$K_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{matrix} \cdots \text{R1} \\ \cdots \text{R2} \\ \cdots \text{R3} \\ \cdots \text{R4} \end{matrix} \quad \text{or} \quad K_2 = \begin{bmatrix} -1 & 0 \\ -1 & -1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix} \begin{matrix} \cdots \text{R1} \\ \cdots \text{R2} \\ \cdots \text{R3} \\ \cdots \text{R4} \end{matrix}$$

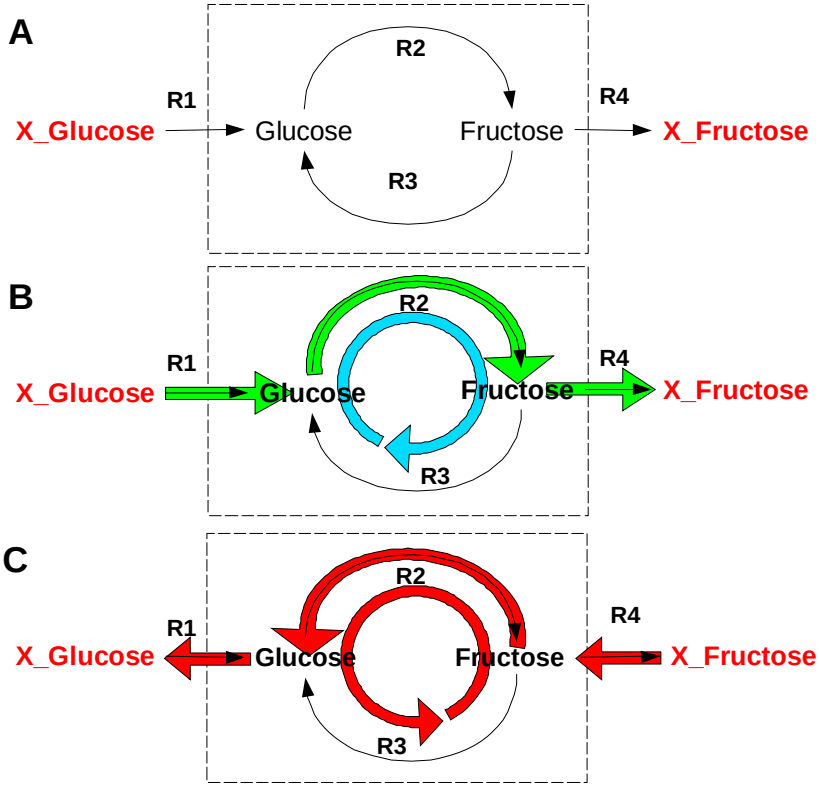


Figure 2-3 Using the null space to define network flux
A) A cyclic model system, B) Routes (fluxes) from the null vectors (column flux vectors) that comply the reaction reversibility criteria, C) Fluxes that do not comply with the reaction reversibility criteria.

The first null space K_1, does respect the reaction's reversibility (Figure 2-3 B) but the second null space K_2, does not comply with the reaction's reversibility as indicated by the sign of the elements in K_2 (Figure 2-3-C). Even though the latter is a valid solution of the system, it is not a practically applicable solution to metabolic model in Figure 2-3 A, as the direction of the flux going through the reactions does not follow the irreversibility criteria giving an impractical solution.

If the null space is block-diagonalisable, then it is possible to identify subnetworks or independent components of the system which are either completely disconnected or whose steady state fluxes are independent from the fluxes from the rest of the network (Heinrich & Schuster, 1996). Schuster's group wrote a program 'BlockDiag', to calculate the block-diagonalizable form of the null space.

The use of the block-diagonalizable form to predict the subnetworks (independent components) in the system also has problems. For a large system, obtaining the block diagonalizable form of the null space is a computationally difficult task. Despite these shortcomings, the null space is important to obtain the condensed form of stoichiometry matrix (explained in detail in the next section) of the system. The null space is also important for identification of functionally correlated reactions (also known as enzyme subsets) of the system, and it is discussed in Section 2.3.

2.3 Structural modelling

Analysis of the structural models can be carried out using various methods. The structural modelling techniques are based on the assumption of the steady state. As illustrated in Figure 1-1, various techniques were developed for analysing structural metabolic networks. The present work was carried out using two major aspects of structural modelling:

- Enzyme subsets are groups of enzymes that always operate in fixed flux proportions at a steady state (Pfeiffer *et al.*, 1999).
- Elementary modes are a group of minimal sets of reactions that can operate at steady state with all irreversible reactions proceeding in the appropriate directions (Schuster & Hilgetag, 1994; Schuster *et al.*, 1994).

Other aspects of the structural modelling include conservation relationships, dead enzymes, orphan and dead-end metabolites and futile cycles.

2.3.1 **Conservation relationship**

Conservation relationships represent constraints on conserved moieties in the system. Conserved moieties are molecular subgroups (chemical moieties) which always remain in the system without loss of integrity (Heinrich & Schuster, 1996). Typical examples of conservation relationships are ATP and ADP, and NAD^+ and NADH. In the metabolic system, the total amount or concentration of the adenosine moiety in ATP and ADP remains constant (i.e. [ATP] + [ADP] = constant), provided there is no generation or degradation of the adenosine moiety as shown in the model system in Figure 2-4.

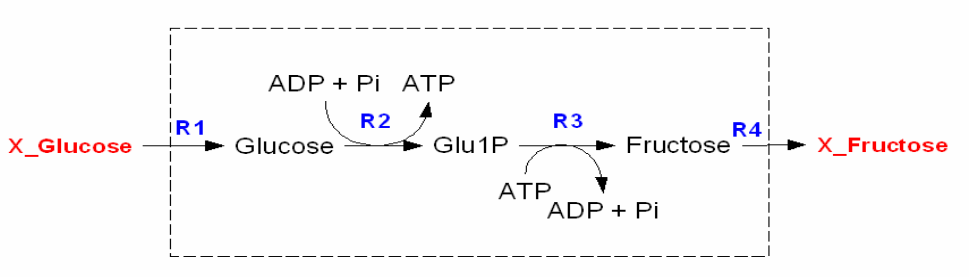


Figure 2-4 Example of the conserved moiety in a model system

Mathematically, such pairs can be easily identified from *N*; all elements in the row for ATP and ADP of *N* are exactly the same, except that all elements in one of the two rows are multiplied by -1. Such rows are called linearly dependent rows of the matrix. Conserved moieties in the network are identified as linear dependencies in the rows of the stoichiometry matrix and can be calculated from the stoichiometry matrix ***N*** of the network (Schuster & Hilgetag, 1995).

In the conservation moiety matrix (Γ), each row relates to a particular conserved sum. The number of rows indicates the number of linearly independent conserved moieties in the network. Elements in a particular row indicate which metabolites contribute to a particular conserved cycle. Such cycles were believed to be conserved during the evolution of a network since the total amount of a particular moiety in a network remains time-invariant.

$$N^T \Gamma = 0 \qquad (6)$$

where $\boldsymbol{N}^T$ is a transpose of $\boldsymbol{N}$, $\boldsymbol{\Gamma}$ represents the conservation moiety matrix, and 0 is a zero vector.

Mathematically, the conservation moiety matrix can be obtained from equation 6 or from the left null space of $\boldsymbol{N}$ (Famili & Palsson, 2003) using equation,

$$\Gamma^T N = 0^T \qquad (7)$$

where Γ^T is a transpose of conservation moiety matrix (or the left null space of $\boldsymbol{N}$) and 0^T is a zero vector. A recent review by Sauro & Ingalls (2004) gives various mathematical and computational techniques used for obtaining the conservation moiety matrix. The Γ can obtained by the Gauss-Jordan method (the left null space of $\boldsymbol{N}$). Sauro & Ingalls (2004) used Singular Value Decomposition to obtain Γ for a large metabolic system.

Conservation relationships play a major role in shrinking the kinetic behaviour of a metabolic system. For example, in a system, the total concentration of the conserved moieties remain unchanged, thus reducing the number of variables over time (Heinrich & Schuster, 1996). Recently, Imieliski *et al.* (2006) used a conservation relationship study of a genome-scale metabolic network of *E. coli* to develop a novel growth media.

2.3.2 **Enzyme subsets**

In a metabolic network, some reactions always operate together. A group of such reaction(s) is referred to as an enzyme subset. The concept of an enzyme subset reported by Pfeiffer *et al.* (1999) for metabolic modelling was similar to

the earlier 'monofunctional unit'[3] concept in metabolic control analysis (Kholodenko *et al.*, 1995; Rohwer *et al.*, 1996).

An enzyme subset is defined as a group of enzymes that,

a) Carry flux in a fixed proportion at a steady state, and

b) Only one independent metabolic flux goes through the subset.

The mathematical explanation for the enzyme subsets' calculation is well reported in literature (Klamt *et al.*, 2002a; Pfeiffer *et al.*, 1999). For calculating the enzyme subsets, the null space matrix (K) is used to find the proportional elements of rows (i.e. reactions). All the proportional rows are placed in the same subset. As explained earlier, the null space matrix is not unique; however, the subsets obtained for a system are unique for a given stoichiometry matrix of the system.

For example, in the case of the system shown in Figure 2-3, other possible null space bases can be given as:

$$K_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{matrix} \cdots \text{R1} \\ \cdots \text{R2} \\ \cdots \text{R3} \\ \cdots \text{R4} \end{matrix} \quad \text{or} \quad K_2 = \begin{bmatrix} -1 & 0 \\ -1 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix} \begin{matrix} \cdots \text{R1} \\ \cdots \text{R2} \\ \cdots \text{R3} \\ \cdots \text{R4} \end{matrix} \quad \text{or } K_3 = \begin{bmatrix} -1 & 0 \\ -1 & -1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix} \begin{matrix} \cdots \text{R1} \\ \cdots \text{R2} \\ \cdots \text{R3} \\ \cdots \text{R4} \end{matrix}$$

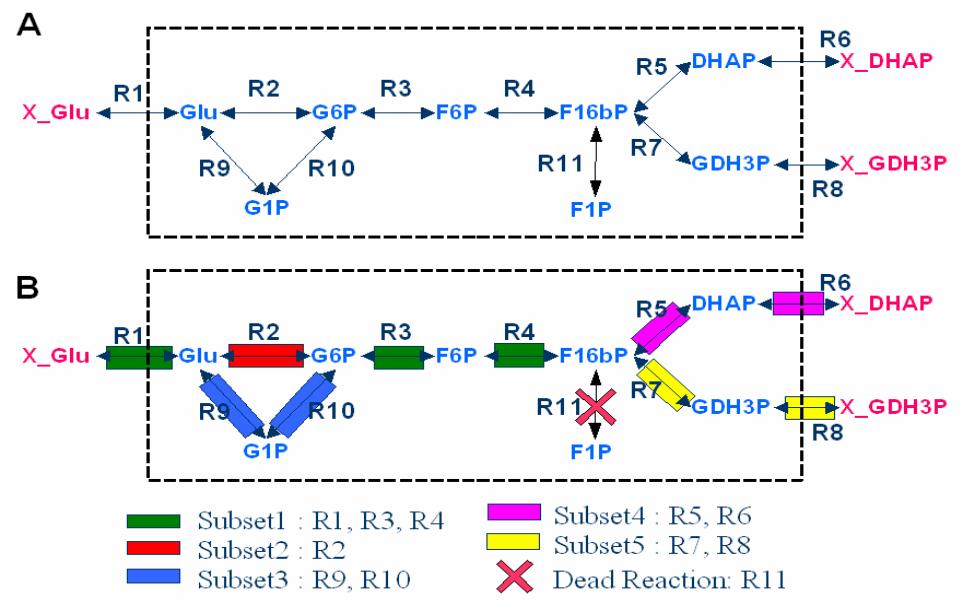
Despite the different possible null space bases, the enzyme subsets of the system remain unchanged. Reactions R1 and R4 belong to a subset, while reactions R2 and R3 belong to two different individual subsets of one single reaction. Thus the enzyme subsets computation is unique even though the basis of the null space is not unique.

[3] In metabolic control analysis, the similar concept of 'monofunctional unit' (super enzyme) was reported by Kholodenko *et al.* (1995) and Rohwer *et al.* (1996), where for all the enzymes in the group a) only one independent flux passed through them, b) no conservation moieties are present in such reactions, and c) no other allosteric or regulatory effects of metabolites present in such reactions or unit.

An enzyme subset can be classified according to the number of reactions it contains. Enzyme subsets can be further classified on the basis of the number of reactions in each subset as:

a) Trivial subsets containing only one reaction;

b) Subsets containing two or more reactions.



C

null space (K)	Rearranging the rows of K
R1 [-1, -1, 0]	**R1 [-1, -1, 0]**
R2 [0, 0, -1]	**R3 [-1, -1, 0]**
R3 [-1, -1, 0]	**R4 [-1, -1, 0]**
R4 [-1, -1, 0]	**R2 [0, 0,-1]**
R5 [0, -1, 0]	**R9 [-1, -1, 1]**
R6 [0, -1, 0]	**R10 [-1, -1, 1]**
R7 [-1, 0, 0]	**R5 [0, -1, 0]**
R8 [-1, 0, 0]	**R6 [0, -1, 0]**
R9 [-1, -1, 1]	**R7 [-1, 0, 0]**
R10[-1, -1, 1]	**R8 [-1, 0, 0]**
R11[0, 0, 0]	**R11 [0, 0, 0]**

Figure 2-5 Enzyme subset for a branched metabolic model system

A. A model branched system, **B**. Subsets of the branched system, **C**. Mathematics of subset computation, rows with proportionality of the elements with the other rows are shown by same colour, form an enzyme subset.

For a linear system similar to the one shown in Figure 2-2A, the entire system forms only one enzyme subset as there is only one independent flux going through all four reactions. However, if a more complex (branched) system such as shown in Figure 2-5-A is considered, the enzyme subset prediction needs a

computational algorithm. For this branched system, six different enzyme subsets are observed.

The mathematics behind the computation of the enzyme subsets for the system in Figure 2-5A can be explained using the null space of the system. Figure 2-5C, shows the null space (kernel) of the system. Each row represents one reaction in the system and each column represents the possible solution space. When the row vectors of the null space matrix are rearranged such that any two row vectors of same proportionality with respect to the corresponding elements (as shown in Figure 2-5C same colour row vectors) are grouped together to represent an enzyme subset, i.e. the ratio of the fluxes of those reactions from all the three solution vectors remains the same for reactions in a subset (as shown in Figure 2-5 B). For the above branched system, six distinct subsets can be identified, (e.g. for R1, R3 and R4, the ratio of the fluxes for any two reactions in all the three solutions is 1:1:1, thus R1, R3 and R4 form an enzyme subset in the system, while R11 has all elements as zero representing a dead reaction). In comparison to the previously discussed (Figure 2-2A) linear system, due to the branching after R1 and R2 in this system (Figure 2-5), reactions R1, R3 and R4 forms an enzyme subset; and reaction R2 forms an independent enzyme subset. Although R1 and R3 must carry the same flux, the flux between them can be shared by R2 and R9 plus R10. Since there is no constraint on the relative proportion in the alternative routes (one from R1 to R4 and another via R9, R10), there is no constant relationship between flux of the other reactions (i.e. R1, R3 or R9, R10) and flux through R2, resulting in R2 as an independent single reaction enzyme subset.

A general outline of the algorithm (Heinrich & Schuster, 1996) for enzyme subset identification is as follows,

- Detect all row vectors of K that are null vectors (i.e. Dead reactions)
- Normalise each of the remaining row vectors of K by dividing by its greatest common divisor.
- Compare any normalised row vector with any other. If they are the same and there are no contradictions in the directionalities of irreversible reactions, the corresponding reactions belong to the same subset. If there is

a problem with reversibility, break the reactions into subsets with respect to reversibility criteria.

An important feature of enzyme subsets is that reactions in a subset can be combined for the simplification of the model. Reactions in the enzyme subsets can be replaced by a single overall lumped reaction (simple addition of each reaction to give one single lumped reaction) without affecting the model structure.

Suggested roles *in vivo* for enzyme subsets are that they might be functional units in terms of regulation of metabolism, performing coordinated metabolic regulation. Schuster *et al.* (2002b) found a correlation between relative change in gene expression upon the diauxic shift in yeast and enzyme subsets of yeast central metabolism. Reed *et al.* (2004) investigated an *E.coli* genome-scale model. They found that sets of reactions that are always used together in optimal solutions (i.e. enzyme subsets) showed moderate agreement with the currently known transcriptional regulatory structure in *E.coli* and available expression data. Our findings on the genome structure and enzyme subsets are discussed in chapter 4.

Condensed stoichiometry matrix

As discussed above, one advantage of using enzyme subsets is that one can lump together the reactions in the same enzyme subset. The condensed stoichiometry matrix is a modified stoichiometry matrix of all the lumped reactions from all the enzyme subsets (Pfeiffer *et al.*, 1999). This not only reduces the number of reactions (i.e. variables) of the system, but also reduces the number of metabolites internal to subsets, giving a more condensed form of the stoichiometry matrix of the system. The advantage of obtaining the condensed stoichiometry matrix is that it can speed up the computation of elementary modes (discussed later in this section) for a large system. This is discussed in the next subsection on elementary modes.

2.3.3 **Dead enzymes**

Dead enzymes (strictly detailed balanced reactions (Schuster & Schuster, 1991) or blocked reactions (Burgard *et al.*, 2004)) are those which do not carry any net flux through them at steady state. Such dead reactions are identified from the null space matrix. A row of zeroes in *K* means there can be no net steady state flux through the corresponding reactions in the network, and that such reactions are grouped together and called dead enzymes.

The identification of dead reactions is an important step in model interrogation. Such reactions might provide some insights on the incompleteness of the biological system. Dead reactions may signify errors/omissions in the metabolic reconstruction (wrong reaction stoichiometry or missing experimental data in the metabolic network). It might also be possible that a reaction or enzyme is actually inactive in the metabolism at the given physiological conditions of the system (e.g. for a system based on the growth conditions, it is possible to find few catabolic reactions involving secondary metabolites to be inactive). In addition, reactions that maintain equilibrium between conserved metabolites (e.g. adenylate kinase) will be active but have no net fluxes at steady state and may appear as dead. A more detailed discussion on dead reactions and reconstruction of metabolic networks is included in Chapter 3.

2.3.4 **Orphan and dead-end metabolites**

Orphan metabolites[4] appear only once in the metabolic model. Such metabolites are the major cause of the reaction being dead in the system. The identification of orphans helps in correcting the model definition. Dead-end metabolite may appear more than once, but are either only produced or utilized in the system. They pose a lesser threat to the system compared to orphans, but identification of dead-end metabolites is difficult as compared to orphans.

[4] The terms orphan and dead-end metabolite are coined by Dr. Poolman (unpublished data) CSM lab, Oxford.

Since all the orphan metabolites are also dead-end metabolites in the network, to avoid confusion, orphan and dead-end metabolites are always reported separately. Each metabolite has a connectivity score, which represents the total number of reactions utilising or producing the respective metabolite in the network. The only difference in orphan and dead-end metabolite is the connectivity score; orphans have a connectivity score of one.

The dead-end metabolites are only produced or only consumed, but never both, they always have a connectivity score higher than one. For identifying the dead-end metabolites that are not orphans, additional information on the reaction reversibility is needed. A more detailed discussion on orphan and dead end metabolites and the reconstruction of a metabolic model is presented in Chapter 3 and 4.

2.3.5 **Elementary mode analysis**

A mode is a set of reactions of a system. At the steady state condition, the relative flux distribution that fulfils for intermediates and follow the sign restriction for irreversible reactions is defined as a mode of a system and was first reported by Leiser & Blum (1987) and Fell (1990). The concept was later extended by Schuster *et al.* (1994) as an elementary mode. Modes are called elementary if they are non-decomposable (Heinrich & Schuster, 1996) i.e. further elimination of any single reaction from the elementary mode can violate the steady state condition.

By definition (Schuster et al., 2002a), a flux mode (M) can be defined as

$$M = \{\bar{v} \in R^r \mid \bar{v} = \lambda \bar{v}^*, \lambda > 0\}$$

where $\bar{v}$ is a vector of reaction rates, r denotes the number of reactions and λ denotes arbitrary real number, R^r denotes the r-dimensional Euclidean space, and $\bar{v}^*$ stands for a vector fulfilling the following two conditions:

1. $\bar{v}^*$ satisfies the (quasi) steady state i.e. satisfies Eq. (4)
2. Follows sign restriction of the reactions. If an irreversible reaction is present in the system, then the sub vector of $\bar{v}^*$ fulfils the inequality.

A flux mode is an elementary mode (EM) if and only if it fulfils the following condition in addition to the above mentioned two conditions.

3. Simplicity (or non-decomposability) of the elementary modes (Schuster *et al.*, 1999) i.e. there is no other non-null flux vector that satisfies both the above conditions 1 and 2; and involves a proper subset of its participating reactions.

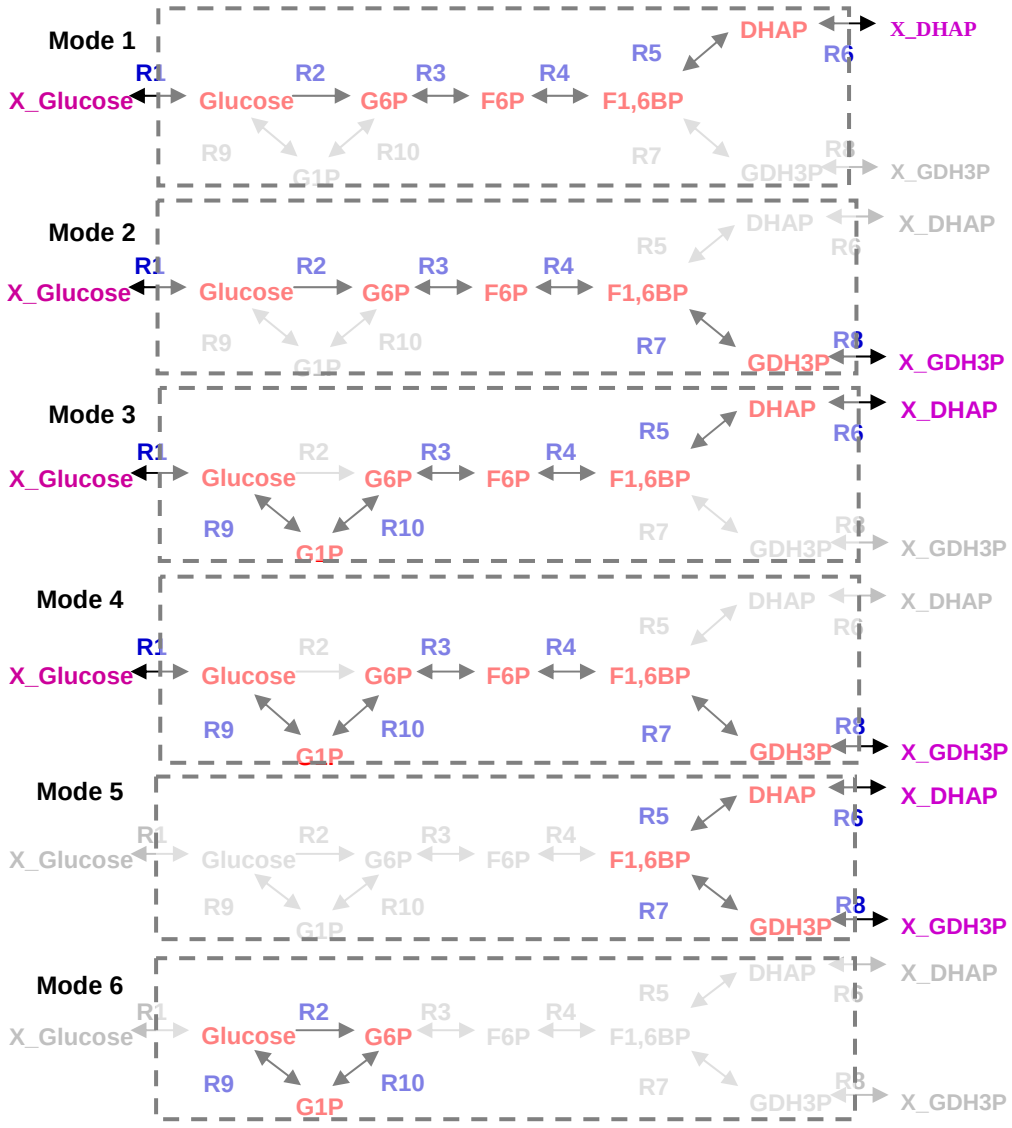


Figure 2-6 Illustration of elementary modes for a model system

Thus, the three important characteristics of elementary modes are, 'minimal' with respect to the utilized reactions, non-decomposable and irreducible. Since an elementary mode cannot be broken down into further elementary modes, one can argue that the study of such modes gives information about the flow of metabolic flux from one external to the other external metabolite. Therefore,

elementary mode can be defined as a fundamental metabolic flux or route for the conversion or production of the external metabolites.

Figure 2-6 shows all possible elementary modes of the metabolic system discussed earlier in Figure 2-5. The system shows six elementary modes, four modes (Mode 1, 2, 3 and 4) utilize external glucose. Mode 5 utilizes one of the products to give another product, while mode 6 forms a cyclic mode called a futile cycle (discussed later in section 2.3.6). Elementary modes can be further classified on the basis of the reaction reversibility constraints, reversible (e.g. modes 3, 4 and 5) modes consist of all reversible reactions while irreversible (e.g. modes 1, 2 and 6) modes consist of at least one irreversible reaction in the system.

Each elementary mode has an overall or net reaction equation with the consumption and production of external metabolites. If external metabolites are neither consumed nor produced, then the elementary mode is an internal or futile cycle (see section 2.3.5). The algorithm for detection of elementary modes was sketched earlier by Schuster and Schuster (1993), Schuster and Hilgetag (1994) and then by Pfeiffer *et al.* (1999) and a mathematical proof was reported by Schuster *et al.* (2002a) who validated and proved that the algorithm exclusively generates all the elementary flux modes of an arbitrary network. Schuster *et al.* (2002a) also suggested an algorithm using the tableau based method of linear algebra to find the unique decomposition of the stoichiometry matrix *N* into elementary modes. Recently another algorithm was suggested by Wagner (2004) and later by Urbanczik and Wagner (2005) which uses null space (K) to find the elementary modes of the system.

Over the last few years, elementary mode analysis has been applied successfully, for instance in bioengineering (Carlson *et al.*, 2002; Carlson & Srienc, 2004a, 2004b; Schuster *et al.*, 2002b), or in functional genomics (for assigning functions to orphan genes). Schuster *et al.* (2002b) reported the use of elementary mode analysis to maximize the synthesis of cyclooctadepsipeptides PF1022 in the fungus *Mycelia sterilia*. Carlson *et al.* (2002) used elementary

mode analysis to investigate the possible effects of biochemical network modifications and altered culturing conditions to optimise poly-β-hydroxybutyrate (PHB) production in recombinant yeast. Poolman *et al.* (2003) investigated and compared dark/light metabolism in a model of Calvin-cycle and oxidative pentose phosphate pathway (OPPP) by means of elementary modes. Carlson and Srienc (2004b) described the prediction of unique pathways in *E.coli* for efficient conversion of glucose and O_2 in to new cells and the maintenance of energy under different growth conditions. In addition, they also reported a close agreement between predicted and experimental observations on all the metabolic fluxes.

Elementary modes help in the understanding of the overall metabolism by providing significant information about all the possible routes from one external metabolite to the other external metabolites (from source to sink) at steady state. Therefore, they could play an important role in understanding the enhancement of the product (of commercial importance) and in determining how the modes are utilised for the production of such a product in the systems under study. The decomposition of a reaction network into elementary flux modes provides a complete list of all the physiological functions of the network revealing the diversity and functional richness of a network (Heinrich & Schuster, 1996). To investigate the diversity and functional richness, the number of modes employing a certain reaction can be calculated to give a relative frequency of a reaction that might indicate the importance or even essentiality of an enzymatic reaction, or demonstrate the requirement for certain transporters in enabling various metabolic pathways. This principle was applied further to 'damage analyses in the present work, and is discussed later in chapter 6. Moreover, the incidence of elementary modes with identical overall net reactions is a measure of network robustness (Stelling *et al.*, 2002).

All flux distributions of a metabolic network that are possible at steady state are linear combinations of the fluxes carried by elementary modes. Poolman *et al.* (2004b) reported the appropriate assignment of actual flux values to the elementary modes. However, in larger networks, Klamt and Stelling (2002b)

showed that the calculation of elementary modes was hampered by the combinatorial explosion of the possible routes. They observed that for an *E. coli* model with just four source metabolites (110 reactions and 88 metabolites), 507632 elementary modes are possible. The in-depth analysis of each mode from such large number is practically impossible.

For genome scale large metabolic networks, computation of elementary modes is a computationally hard problem and might need high performance computation. One of the challenges is not just the computation of elementary modes, but also the identification of the physiologically significant modes from all possible theoretical elementary modes.

2.3.6 Substrate cycles or futile cycles

As mentioned in an earlier section, elementary modes with no consumption or production of external metabolites are called internal or substrate cycles. An elementary mode starting from one internal metabolite and producing the same metabolite via (an)other metabolite(s) using two or more simultaneously active reactions in the system is an example of a futile mode, suggesting a wasteful route in the metabolic network. Earlier such cycles were considered as imperfect and thought to be unnecessary and wasteful routes and hence were given the name 'futile cycles'.

In the metabolic network, some reaction sequences appear as cyclic flux or modes. Such cyclic routes can be either essential cyclic sequences (e.g. TCA cycle), or non-essential sequences. The essential cyclic routes may not be necessarily futile cycles, since their net production is not 'zero'. In the latter case, the cyclic route is observed only if all the reactions are simultaneously active at the same time. Such cycles may not be traced easily on a metabolic chart (Stephanopoulos *et al.*, 1998).

In Figure 2-3, reaction R2 converts glucose to fructose and R3 converts fructose back in to glucose. If both the reactions are active simultaneously, then there can be a cyclic flux converting all the glucose to fructose (*via* R2) and then back to

glucose (*via* R2). Such cyclic flux generally appears to consume ATP (e.g. phosphorylation of ATP) and performs an important role in energy dissipation without any change in the net flux. Such cycles with the occurrence of two oppositely directed sets of reactions that would operate to achieve no change other than dissipation of energy are called 'substrate cycles'(Fell, 1993, 1997)

Fell (1997) gave a modified and generalised definition of internal substrate cycles as:

a) The flux pattern in the network cannot be fully described as a combination of the minimum number of linear paths needed to account for the mass flows connecting the source and sink metabolites.
b) One of the additional fluxes needed to complete the description is a feasible, internal cyclic route.
c) If one step of the cycle is deleted, the network still remains functionally capable of connecting the observed input fluxes to the observed output fluxes.

Mathematically, identification of futile cycles needs an analysis of elementary modes. Modes with no involvement of external metabolites and zero net stoichiometry form internal cycles. The identification of futile cycles gives information about avoiding wasted fluxes following the over expression of enzymes (Leiser & Blum, 1987; Schuster *et al.*, 1999).

2.4 Other approaches to structural modelling

Other approaches used for analysis of the stoichiometry matrix include linear programming based approaches such as extreme pathway analysis, metabolic flux coupling, flux balance analysis as well as metabolic flux analysis. These techniques are briefly discussed in this section.

2.4.1 Metabolic flux analysis

The main aim of metabolic flux analysis (MFA) is to determine the flux distribution (v) in a metabolic network from experimental data in a metabolic network in steady state. In MFA, the possible solution space is reduced by measuring some of the reaction rates (mostly source metabolites uptake or sink

metabolite excretion rate) (Stephanopoulos *et al.*, 1998). These measured or known rates are used for calculation of other unknown rates.

Starting from equation 4, N and ν are partitioned into the known or measured (m) part and unknown (u) part by rearranging the columns in N and components of vector ν (Klamt *et al.*, 2003).

$$N\nu = 0 = N_u\nu_u + N_m\nu_m$$
$$ie.\ N_u\nu_u = -(N_m\nu_m) \qquad (8)$$

The ideal, unique and exact solution occurs only if N_u is a square matrix and invertible, allowing all unknown rates in ν_u to be determined. In a general case, the rank of N_u (rank(N_u)) determines redundancy (can be redundant or non-redundant) and determinacy (determined or underdetermined) (Klamt *et al.*, 2002a). Redundant systems are identified by checking the inconsistencies (gross measurement errors or modelling errors). For underdetermined systems, only a few elements of ν are uniquely calculable (and can be found by the null space analysis of N) (Van der Heijden *et al.*, 1994).

In general, MFA is a non-optimisation based method used to analyse specific reaction fluxes in a small network (Avignone-Rossa *et al.*, 2002). In the case of large systems, Wiechert (2001) used ^{13}C labelling to assign rate measurement with MFA to calculate all reaction rates.

2.4.2 **Flux balance analysis**

Flux balance analysis (FBA) is an optimisation based method (Varma & Palsson, 1994). An objective function is optimised according to given constraints. FBA also uses the quasi steady state assumption (equation 4) with additional constraints such as optimal function of network and reaction capacities over the null space analysis. The characteristic assumption of FBA is based on the following three constraints:

- Optimal function of a network such as maximizing growth (or maximizing the product yield);
- The quasi steady state assumption of system; and

- Reaction capacities (such as irreversibility of reaction) (Edwards *et al.*, 2001) which can be represented and solved as a linear optimization problem.

The constraints can be formulated as:

$$N_{full}\,\bar{v} = \bar{b} \qquad (9)$$

where N_{full} is a full stoichiometry matrix, which includes both internal as well as external metabolites of the system. Vector $\bar{b}$ represents the rate of the amount of uptake or excretion of external metabolites, and the rate of change of internal metabolites is zero. The full stoichiometry matrix is partitioned into an upper part with all the external metabolites, and a lower part with all the internal metabolites. Vector $\bar{b}$ is also partitioned into an upper part with the values of the known uptake or excretion rates of external metabolites, and a lower part with 'zeros' for the corresponding internal metabolites. In fact, Equation 9 is considered as the most general form of the basic equation described earlier as Equation 4.

The objective function can be given as:

$$\alpha_i \le v_i \le \beta_i$$
$$c^T r = c_1 r_1 + c_2 r_2 + \cdots + c_q r_q = \max \qquad (10)$$

Where, α_i and β_i denote the lower and upper boundaries for the reaction rates, and vector C represents the optimization criteria such as weights of the reaction rates.

FBA accounts for reaction reversibility *via* α_i. For an irreversible reaction *i*, the lower boundary is set to zero (i.e. α_i =0) which cuts half the dimension of the null space. Other known boundaries are set to uptake rates or excretion rates. Unknown boundaries remain unconstrained and assigned as infinity. For maximizing the growth rate, for example, only the coefficient corresponding to the growth rate is set to '1', and all the others to '0'. Additionally, MFA can also be applied to FBA (Eq. 2.8), if some measurements on reaction rates are available (Stephanopoulos *et al.*, 1998).

One of the advantages of FBA is its yield prediction capability for products of commercially important microorganisms (Nielsen, 1998; Varma *et al.*, 1993).

Ibarra *et al.* (2002) have shown bacteria such as *E. coli* to behave (stoichiometrically) optimally with respect to biomass yield under selective pressure, and this optimal behaviour can be quantified *in silico* (Edwards *et al.*, 2001). Another very useful application of FBA is in predicting functionality of the reactions of the system after a certain gene deletion (Edwards & Palsson, 2000). Reactions are eliminated from the network *via* gene deletion, and optimization of the new network is then performed.

The main criticism of FBA is that the optimality criterion is not always applicable. Not all the cells, especially bacteria, behave stoichiometrically optimally under all circumstances. In addition, the calculated optimal flux distribution is not always unique. Even though the optimal value of the objective function is unique, the calculated optimal flux distribution itself is not unique and infinitely many solutions are possible. Since only one particular solution is given by FBA, only the (optimal) input/output-relationship is always uniquely determined, and no complete predictions can be made about the internal flux distribution in the system.

a) Minimization of Metabolic Adjustment (MoMA)

MoMA is a further extension of the FBA approach (Segre *et al.*, 2002). It is based on the assumption that in response to a mutation the system searches for the nearest solution to the wild-type metabolism in the new feasible space of steady state flux distributions (which is part of the wild type solution space). Thus, the mutant may not behave optimally with respect to the theoretical solution for its actual network resources. The mutant can adjust its metabolism with minimal effort as compared to the wild type strain. Segre *et al.* (2002) showed this approach to lead to better predictions than FBA.

b) Energy balance analysis (EBA)

In EBA, additional constraints based on the thermodynamic properties of the network are applied in addition to FBA (Beard *et al.*, 2002).

2.4.3 **Extreme pathway analysis**

In the linear programming approach (Varma & Palsson, 1994), the metabolic steady state is considered with a flux balance equation as given in the earlier subsection $N_{full}v = b$ (see Equation 9 for further details). The above equation is underdetermined, since the number of fluxes normally exceeds the number of metabolites, giving more than one solution. Palsson's approach uses a linear optimization method by stating an objective and seeking its maximal value within the stoichiometrically defined domain.

Extreme pathway (EP) analysis (Schilling *et al.*, 1999) uses a similar approach to elementary mode analysis. Based on system stoichiometry and limited thermodynamics, a flux cone is derived using convex analysis. The edges of the steady-state flux cone can be used to represent any flux distribution achievable by the metabolic network (Schilling *et al.*, 2000).

An algorithm has been presented to determine the set of extreme pathways for a system of any complexity, and a classification scheme was introduced for the characterisation of these pathways (Papin *et al.*, 2003).

2.4.4 **Flux coupling analysis**

Burgard *et al.* defined another concept of co-sets or flux coupled reactions in a system, a group of reactions with a similar pattern of flux flowing through the reactions (Burgard *et al.*, 2004).

In this approach of flux coupling, they determined the relationship between any two metabolic fluxes, v_1 and v_2. These could be-

- **Directionally coupled** if a non-zero flux for v_1 implies a non-zero flux for v_2 but not necessarily the reverse;
- **Partially coupled**, if a non-zero flux for v_1 implies a non-zero, though variable, flux for v_2 and vice versa; or
- **Fully coupled**, if a non-zero flux for v_1 implies not only a non-zero but also a fixed flux for v_2 and vice versa.

It was observed that enzyme subset concept (Pfeiffer *et al.*, 1999) is the same as the fully coupled sets in the system whereas dead reaction subset is called as a set of blocked reactions.

Blocked Reactions

Another type of reaction set reported by Burgard *et al*. was blocked reactions. These are reactions that do not carry any flux in the metabolic network. These reactions were discussed in the earlier section as dead reactions, or strictly balanced reactions under steady state metabolic networks. Analysis of blocked reactions by Burgard *et al.* (2004) suggested following possible reasons for a reaction being dead:

- Steady-state assumption,
- Imposed uptake/secretion scenarios,
- Growth requirements, and
- Energy production requirements

2.4.5 Graph theoretic analysis

The applications of graph theoretical approaches to structural metabolic networks are quite well known. The applications of graph theoretical approaches include

a) Graph theoretic damage analysis;

b) Minimal cut sets analysis.

Both approaches are discussed in detail in Chapter 6.

2.5 Software for structural metabolic modelling

For structural modelling of metabolic networks there are a number of software packages available. In the following section, the most commonly used software packages are discussed.

2.5.1 Software packages for structural modelling

Gepasi[5] (Mendes, 1993) was originally developed for the kinetic modelling of the biochemical reaction networks, can be used for analysis of structural

[5] www.gepasi.org/

properties of kinetic models. Gepasi can be used for structural modelling of small metabolic networks. Copasi[6] is the successor of Gepasi, rewritten with a full front-end development in QT toolkit and available only for the Windows operating system.

Metatool[7] (Pfeiffer *et al.*, 1999) was the first software dedicated to structural modelling of the metabolic network. It is open source[8] software written in 'C'. Metatool uses its own textual model file specification and returns results in a single text output file. Its output consists of a list of enzyme subsets, elementary modes and conservation moieties. It also has additional modules or programs for structural modelling. These are:

- 'BlockDiag' is a program code for computing and block-diagonalising the null space matrix from the stoichiometry matrix of a chemical reaction network which is based on the code written in Pascal (Schuster & Schuster, 1991). The block-diagonalisation helps in detecting isolated or disconnected subsystems or parts of the network with independent fluxes.
- 'OptiMode' is a program code for detecting the elementary mode with the highest molar yield for a specified substrate - product pair from the output file of Metatool.
- 'Separator' (Schuster *et al.*, 2002c) is a program for decomposing large biochemical networks into smaller ones based on the degree of connectivity. Degree is the number of neighbours to a node of a graph (also called connectivity.) This helps in avoiding combinatorial explosion of metabolic pathways.
- 'Reducing the number of modes' program can be used to calculate all the elementary modes for the complete variety of external and internal metabolites and provides the approximated minimal number of modes (Dandekar *et al.*, 2003).

[6] www.copasi.org/
[7] http://www.bioinf.mdc-berlin.de/projects/metabolic/metatool/ and http://pinguin.biologie.uni-jena.de/bioinformatik/networks/metatool/metatool5.0/metatool5.0.html
[8] http://www.opensource.org/

phpMetatool[9] is a web based implementation of Metatool allowing the user to run Metatool on a web server *via* web browsers.

Jarnac[10] (Sauro, 2000) was developed for describing and manipulating cellular system models and can be used to describe metabolic, signal transduction and gene networks, or almost any other physical system that can be described in terms of a network and associated flows. It is the successor to another metabolic modelling software SCAMP (Sauro, 1993; Sauro & Fell, 1991) developed at this university (Sauro, 1986). It is written in a simple control language called Delphi (Pascal). It also has a command line support where user-defined functions and external modules are executed for model interrogation. Jarnac also provides computation of structural properties such as null space, elementary modes, enzyme subsets, conservation moieties and futile cycles.

FluxAnalyzer[11] (Klamt *et al.*, 2003) uses commercial program MATLAB, and provides various tools for stoichiometric analysis of metabolic networks. Interactive flux maps can be used to realize the core concept of visualization and the abstract network model can be linked with network graphics. It also provides powerful collection of tools and algorithms that are available within Matlab. It provides menu-controlled tool bars such as a toolbar for metabolic flux analysis, flux optimization, detection of topological features and pathway analysis by elementary flux modes and minimal cut sets. CellNetAnalyzer is the successor of the FluxAnalyzer.

Yana[12] (Schwarz *et al.*, 2005) is a successor of Metatool, written in Java, for the structural analysis of metabolic networks. It performs elementary mode analysis and incorporates new algorithms for mapping elementary mode activities to enzyme activities and *vice versa*. Yana is available for Windows and UNIX operating systems. It includes a graphical user interface for the easy editing of

[9] http://www-bm.ipk-gatersleben.de/tools/phpMetatool/
[10] http://sbw.kgi.edu/software/jarnac.htm
[11] http://www.mpi-magdeburg.mpg.de/projects/cna/cna.html
[12] http://www.biozentrum.uni-wuerzburg.de/yana.html

metabolic networks and is capable of reading and writing the Metatool file format as well as the SBML[13] Level 2 file format.

PySCeS[14] (Olivier *et al.*, 2005) is quite similar to ScrumPy, as it is also written in Python, but developed for kinetic modelling with some additional support for structural modelling. PySCeS can be installed on both Windows and UNIX platform.

2.5.2 **ScrumPy**

ScrumPy[15] is a metabolic modelling package written in Python in the Cell Systems Modelling Group, Oxford (Poolman, 2006a). The features that make ScrumPy different from other software used for structural modelling of metabolisms are as follows:
Apart from Jarnac and FluxAnalyzer, not all the software packages give much flexibility for the interrogation of the model to the user. Jarnac gives a command line support option to the user.

A software package is developed by a developer (one who develops software tools or packages) for others. After developing a metabolic modelling toolbox (software), one can divide end users into a user (one who just needs to analyse his system using the already well developed methods (e.g. elementary modes, enzyme subsets etc.) and a programmer (developer) who wants to use the above existing methods for further development of new methods.
The selection of the metabolic modelling software depends on the identity (user/developer or both) and the needs of the end user. However, a choice of the flexible software is always preferred for research purposes. A user may need to perform various tasks similar to a programmer, and may need to extend the present tools into new tools for his research. To tackle such critical issues, ScrumPy was developed for both kinetic as well as structural modelling.

[13] http://www.sbml.org
[14] http://pysces.sourceforge.net/
[15] http://mudshark.brookes.ac.uk/ScrumPy/

The advantages of choosing Python (Lutz, 2001; Lutz & Ascher, 1999) as a programming language for ScrumPy are elaborated as follows:

- Python is a simple, modular, easy to code, high level programming language. Python is much easier to learn (as a user) as compared to other high-level programming languages.

- Since it is an interpreter based language, Python code gives an error message but it never crashes.

- In addition, Python's modularity makes it simpler and easy to write small pieces of independent codes, which help to manage, maintain and modify such small codes for future use.

- Other advantages of Python are, its object orientation (supports packages, modules, classes, objects, libraries etc.)

- It is platform independent and an open source programming language.

- Python bridges the gap between C and shell programming. It also has a good C interface *via* Swig library.

- Graphical User Interface based applications can be developed using 'Tkinter' library, making the application simple from complex commands for the user.

- Python also has a library of statistical, numerical and scientific tools (e.g. NumPy, Numarray, SciPy, statpy etc.) for powerful mathematical and scientific programming methods.

- For biologists, Python has the BioPython library, which contains many tools for computational molecular biology such as Blast, Fasta, phylogenetic analysis, cluster analysis etc.

All the above tools are also available in ScrumPy for further efficient analysis of the system.

2.5.3 **ScrumPy in brief**

ScrumPy metabolic model files (Figure 2-7) are plain text files with a list of reactions names, reaction stoichiometry with metabolites and reversibility. Some details are as follows:

- Any line starting with hash ('#') is a comment for ScrumPy software.
- Directives include special instructions which ScrumPy follows while loading model file. E.g. Structural() directive loads model file only for modelling of structural aspects of metabolic network.
- Reaction names always end with a colon (':') and are always followed by a stoichiometric reaction equation. If a reaction is reversible, then '<>' sign is included, else '->' represents an irreversible reaction.
- The '~' after the reaction equation indicates that the reaction is assigned a default rate kinetics, as the model is analysed for structural modelling only and not for kinetic modelling. In case of kinetic models, the reaction stoichiometry is followed by reaction kinetic data.

A manual for details on the model file specifications in ScrumPy (Poolman, 2006b) can be found online from http://mudshark.brooeks.ac.uk/ScrumPy/. ScrumPy converts the model file into a python objects called 'model'. The model object has various methods and attributes such as stoichiometry matrix, enzyme subsets, elementary modes, etc., for further analysis of the model. If a reaction contains the same metabolite on both sides, then during the stoichiometry matrix conversion, the coefficient of the left side metabolite (i.e. metabolites or substrates of a reaction that will be utilized) will be subtracted from the coefficient of the same metabolite on the right hand side, resulting in the wrong description of the reaction in the stoichiometry matrix.

e.g. Rx1: Glucose + ATP <> Glucose + ADP + Pi

```
## Bhushan Bonde
## Date: Nov15 2003
## Model generated from
## DFell Data for E.coli metabolism

Structural()
ElType(float)
External(Gluc, CO2, O2, NH3)
External(FAD, FADH2, Q, QH2)

#List of Reactions

PhosphoTf_system_ir:
        Gluc + PEP  -> PYR + G6P  ~
Phosphoglucose_I_r:
        G6P <> F6P  ~
PhosphofructoK_ir:
        F6P  +  ATP  ->  ADP  +  F16DP  ~
Fructose16_biphosphatase_ir:
        F16DP -> F6P  +  Pi    ~
Fructose16_biphosphate_aldolase_r:
        F16DP <> G3P  + DHKP  ~
```

Comments from user

Directive for ScrumPy model execution, model definitions etc.

Unique metabolic reaction ID followed by reaction stoichiometry Additional information about the enzyme, reaction, gene etc can be provided as a comment after each reaction

Figure 2-7 Specifications of ScrumPy metabolic model file format
(Only first few lines in the model file are shown)

ScrumPy (or any other structural modelling software) will produce the following stoichiometry matrix for the above reaction,

$$\begin{matrix} \\ \text{Glucose} \\ \text{ATP} \\ \text{ADP} \\ \text{Pi} \end{matrix} \begin{matrix} Rx1 \\ \begin{bmatrix} -1+1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \end{matrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

The other possibility is that only one side of the metabolite is added to the stoichiometry matrix giving another wrong specification as follows,

$$\begin{matrix} \\ \text{Glucose} \\ \text{ATP} \\ \text{ADP} \\ \text{Pi} \end{matrix} \begin{matrix} R1 \\ \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \end{matrix} \quad \text{or} \quad \begin{matrix} \\ \text{Glucose} \\ \text{ATP} \\ \text{ADP} \\ \text{Pi} \end{matrix} \begin{matrix} R1 \\ \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \end{matrix}$$

All the above stoichiometric descriptions result in further errors. Therefore such reactions should not be included in the model definition, or such reactions should be corrected with the additional information based on the compartment

of the substrates. The problems due to same metabolite on both sides of the reaction are discussed in Chapter 3.

The following methods from ScrumPy were used with additional modules.

- ConsMoieties() uses the Gauss-Jordan algorithm for the calculation of conservation relationships.

- EnzSubsets() is performed by a shortened version of the algorithm explained in section 2.3.2, in which the normalization step is skipped.

- For the computation of elementary modes, ElModes() uses the procedure as provided by (Schuster *et al.*, 2002a).

PyoCyc (Poolman, 2004) is another Python package written by Dr. Poolman for accessing BioCyc databases, and has many facilities to interrogate locally installed BioCyc database. It provides tools for obtaining information on reaction to gene associations, enzyme classes and their reactions that helps in model definition correction. It has many additional features for searching the database for any reaction and/or metabolite. However, the main feature of PyoCyc is the automated building of large metabolic ScrumPy models directly from the various BioCyc databases for specific organisms (e.g. EcoCyc). It generates a large genomic-scale model in only a few seconds for the locally installed BioCyc database for an organism.

2.6 Summery

- The stoichiometric analysis of metabolic networks can be used for the study and understanding of complex metabolic systems. The stoichiometric analysis involves the structural analysis (the 'topology') of the network. This makes stoichiometric constraints-based modelling a good choice as an alternative to kinetic models for analysing large-scale networks.

- The computation of the null space matrix of N is an important step for structural analysis. Another advantage of the null space analysis over the

elementary mode analysis is that the Null space matrix can be computed successfully for a very large stoichiometry matrix on a simple processor computer (more than 1000 reactions), while elementary mode analysis is still difficult for such a large metabolic model.

- The null space (or kernel) analysis of the stoichiometry matrix gives groups of reactions called 'enzyme subsets', which may shed light on the significance of such cluster of reactions in larger living systems. Though the basis of the null space is not unique, the enzyme subsets obtained from any possible null space basis are always the same for a given system.

- Finding the linear dependencies of the rows of stoichiometry matrix aids identification of conservation relationships. Understanding of such conserved moieties may be useful in improvement of the yield of commercially important products or developing new targets such as drugs.

- The study of substrate cycles or futile cycles plays an important role in understanding the metabolic network. It identifies unavoidable wasteful fluxes in the biological system.

- Other stoichiometric analysis methods like Extreme Pathway Analysis (EPA), Flux Balance Analysis (FBA) and Metabolic Flux Analysis (MFA) are suitable if experimental flux values (uptake rates) are known or can be measured. Structural analysis of stoichiometry matrix does not require such information on flux values. With just a given stoichiometry of the system, important information about the conservation relationships, futile cycles, and dead enzymes are obtained from the metabolic network.

Chapter 3
Reconstruction of genomic scale metabolic networks

3.1 Introduction

Traditionally, small metabolic models were built manually for kinetic modelling and simulations. However, a rapid increase in the availability of large genomic and metabolic data has facilitated the reconstruction or representation of the metabolic reactions of a living cell in a large complete model. As indicated in Chapter 2, structural models can be successfully built using stoichiometric information. Building large structural models does not require information on reaction kinetics, but needs information on reaction stoichiometry. Obtaining such information for small, well studied systems such as TCA cycle or glycolysis is quite easy but obtaining similar information for all the metabolic reactions in a single cell is a challenging although not an impossible task. This chapter discusses issues related to the collection, validation and modelling aspects of models based on the metabolic and genomic information.

3.1.1 Reconstruction of genome scale metabolic models

Metabolic model reconstruction is a process of collecting all the relevant data available on the genomic and metabolic context to understand the overall bio-complexity of an organism (Covert *et al.*, 2001). A reconstruction of a model[16] involves collecting all the relevant metabolic information of an organism and then compiling it in a meaningful way so that application of various metabolic tools can be performed on the data. As shown in Figure 3-1, the correlation between genome and metabolism is made by searching various databases, such as KEGG and GeneDB for particular gene, enzyme or protein name. For example, a search can be conducted based on the protein name or the Enzyme Classification number (EC) in order to find the associated gene (Francke *et al.*, 2005). The metabolic network reconstruction and simulation allows a comprehensive insight into the molecular mechanisms of a particular organism,

[16] Here onwards, the term metabolic model will always represent genome scale metabolic networks.

correlating the genome with molecular physiology or various metabolic systems (such as glycolysis, Krebs cycle, and pentose phosphate pathway).

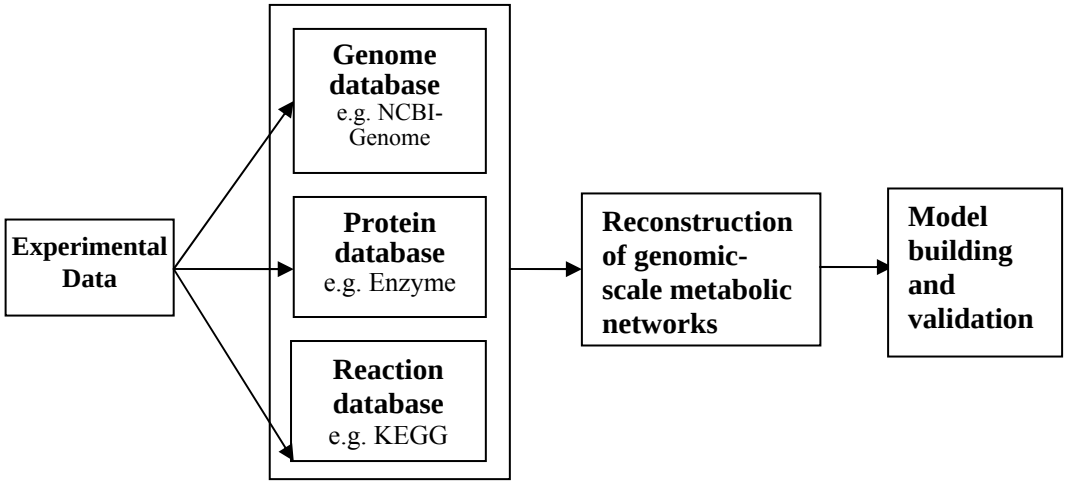


Figure 3-1 Schematics of data driven genomic scale metabolic model reconstruction

3.2 Databases and tools for reconstruction of large metabolic models

3.2.1 Genomic databases

For construction of a genomic scale model, one can start by looking at the genes in the specific organism as given by different databases.

- GenBank[17] provides partial or complete DNA and RNA sequences for more than 165000 organisms (Benson *et al.*, 2005).

- Entrez Genome[18] provides completely sequenced genomes of over 150 organisms, and the number of the completely sequenced organisms is increasing rapidly.

- While information on the presence of genes in the genome is important, additional information on the clustering of genes (operons, COGs) is also available from various databases. Cluster of Orthologous Genes (COG) database[19] (Tatusov *et al.*, 2003) provides information on

[17] http://www.ncbi.nlm.nih.gov/Genbank/index.html
[18] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Genome
[19] http://www.ncbi.nlm.nih.gov/COG/

orthologs/ paralogs by comparing the functionally similar proteins from several completely sequenced genomes (Tatusov *et al.*, 2000).

- The RegulonDB[20] (Salgado *et al.*, 2004) provides information on the predicted *E.coli* operons. ODB[21] database reports both known operons as well as putative or predicted operons in *E. coli* (Okuda *et al.*, 2006).

3.2.2 **Enzyme and protein databases**

Information on the enzymes in a cell can be obtained from databases like BRENDA[22] (Schomburg *et al.*, 2004). This database additionally provides extensive literature-based information on enzymes. It also provides additional information on the kinetic parameters (K_m, V_{max}) reported in the literature for some of the enzymes.

The ENZYME[23] database is based on the recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (IUBMB) and it describes each type of characteristic enzyme for which an EC (Enzyme Commission) number has been provided. This database is of great value for gathering the information on reactions catalysed by a particular enzyme for developing a metabolic model. It currently holds 4579 entries of enzymes (release 38, updates up to 10-Jan-2006) and is available through ExPASy.

3.2.3 **Metabolic databases**

MetaCyc[24] (Karp *et al.*, 2002b; Karp *et al.*, 2000) is a collection of 205 organism's pathway/genome databases while BioCyc is a collection of organism specific genes, metabolic pathways and enzymes from various organisms. The BioCyc (Karp *et al.*, 2005) databases are classified into three tiers as follows:

- Tier 1 databases like EcoCyc (Karp *et al.*, 2005) are manually and intensively curated by experts in respective organisms.

[20] http://regulondb.ccg.unam.mx/index.html
[21] http://odb.kuicr.kyoto-u.ac.jp/
[22] http://www.brenda.uni-koeln.de/
[23] http://www.expasy.ch/enzyme/
[24] http://biocyc.org

- Tier 2 databases are computationally-derived by PathoLogic (Paley & Karp, 2002) but moderately curated manually (e.g. HpyCyc, AgroCyc).
- Tier 3 databases are only computationally-derived by PathoLogic and Pathway Tool software (Karp *et al.*, 2002a).

While the quality of the data in tier 1 databases is quite good, there are still some challenges that need to be tackled to use the data for genome scale metabolic reconstruction (see result and discussion in Chapter 4).

KEGG[25] (Kanehisa *et al.*, 2006) is an excellent suite of databases and associated software, integrating our current knowledge on molecular interaction networks in biological processes. It consists of the PATHWAY database which gives a collection of manually drawn pathway maps of the molecular interaction and reaction networks. GENES, SSDB and KO databases give information about the genes and proteins in a given organism. COMPOUND, DRUG, GLYCAN and REACTION databases give information about the chemical compounds present and reactions for a given species. The current statistics of KEGG databases shows that it has 34,492 pathways generated from 272 reference pathways with 32 eukaryotes, 271 bacteria and 25 archaea. The LIGAND database has (Goto *et al.*, 2002) 13,477 compounds, 2,577 drugs, 11,161 glycans, 6,485 reactions and the BRITE database has 7,998 KO (KEGG Orthology) groups.

Due to the large amount of information, the KEGG database is the first choice for the reconstruction of a metabolic model for many reseachers. However, the KEGG database contains a large number of errors. Recently various groups tried to identify the source and types of errors present in KEGG database (Green & Karp, 2005). During the present research work, our group also identified various possible problems for reconstruction of large metabolic models using databases such as KEGG and BioCyc (Poolman *et al.*, 2006), some of which will be discussed later in Chapter 4.

[25] http://www.genome.jp/kegg/

Another problem is the inconsistency in the database itself. Recently it was observed that extracting/accessing the same information from the KEGG *via* two different interfaces results in different data (Albert Gevorgyan' Unpublished data, work in progress at CSM, Oxford Brookes University.)

Reactome[26] (Joshi-Tope *et al.*, 2005) is a curated, peer-reviewed database which consists of the reaction as a basic unit. The main focus of Reactome is human biochemical processes but it also contains information on a few other organisms.

3.2.4 **Metabolic reconstruction databases**

The WIT ('What Is There') database was first of its kind to be built with the aim of reconstructing the metabolic network from the genome. Unfortunately, for some unknown reasons, after 7 years of service the WIT database was permanently discontinued. However, two new databases, which supersede the WIT database, SEED and PUMA2, can now be used for the same purpose as WIT.

The PUMA2[27] database gives a high throughput comparative and evolutionary analysis of genomes and metabolic networks with a grid computational backend (Maltsev *et al.*, 2006). It currently contains 1,032 prokaryotic and eukaryotic genomes, and allows an interactive analysis of the sequence data using a variety of bioinformatics tools.

The SEED[28] database contains 38 archaeal, 581 bacterial, 562 eukaryal, 1335 viral and 2 environmental genomes[29]. At present, 26 archaeal, 343 bacterial and 29 eukaryal genomes present in the SEED database are almost completely sequenced.

[26] http://www.reactome.org/
[27] http://compbio.mcs.anl.gov/puma2/cgi-bin/index.cgi
[28] http://theseed.uchicago.edu/FIG/index.cgi
[29] Genome DNA samples collected from various mixtures of (single cell) species from extreme environmental conditions, or collected by sail, surveying marine or from terrestrial microbial populations. One example of such genome is sequencing and analysis of samples taken from the Sargasso Sea near Bermuda.

3.3 Approaches for the reconstruction of genomic scale metabolic networks

Traditionally biologists performed experiments on metabolic systems of various microorganisms resulting in a huge amount of data accumulation over the past few decades. The experiments were performed by different laboratories and research groups without a systematic study of all the enzymes in an organism. The collection of all such relevant metabolic data from scientific literature for an organism resulted in the formation of a large metabolic reaction database. Over time, this collection of information resulted into a large pool of data for experimentally favourable microorganisms such as *E.coli* or yeast. With the development in information technology, various databases were created to handle the diverse and huge quantity of data. Such efforts reduced the initial laborious task of collection and management of the raw, uncategorised and diverse biological data for an organism.

However, such information is still random in terms of building a complete snap shot of metabolism of an organism. The lack of systematic study resulted in gaps in the metabolic data, and the reconstruction of a metabolic network for a microorganism posed a new challenge in data integration or investigation. The reconstruction of a metabolic network from various databases involves three different approaches discussed below.

3.3.1 Metabolic reaction centric approach

Traditionally, this was a favoured approach due to the ease of availability of metabolic databases. In this approach, metabolic reaction and enzymes were used to build the metabolic network model (Jeong *et al.*, 2000). This approach was based on the assumption that irrespective of the presence of a gene for an enzyme, all the known reactions present in the metabolic database for the given organism are always present and active in the cell. Such assumptions may not always be correct. It is now well known that for some physiological conditions, the expression of genes (more precisely the transcriptome) depends largely on the external environmental conditions.

3.3.2 **Genome-centric (Top-down) approach**

This approach was used for microorganisms whose genome sequencing was completed and genes were annotated. The annotated genome provides the list of ORF/Genes. Using the bioinformatics techniques such as BLAST and FASTA, few unannotated genes can be assigned to proteins (enzymes) on the basis of the 'best hits' in other species. If a best match for a gene responsible for an enzyme with some threshold 'confidence' was found, the reaction was added in to the model. This approach also used information from COGs database (cluster of orthologous groups) (Tatusov *et al.*, 2003) which involved the phylogenetic classification of the proteins encoded in complete genomes approach. This approach is now becoming more popular due to the development of various bioinformatics tools, which aid in accessing different databases. Kim and Lee (2001) used this approach for the reconstruction of metabolic pathway model and analysis of *Mycoplasma pneumoniae*.

Table 3-1 List of genomic scale metabolic model for various organisms

	Organism	ORF	Genes	Reactions	Metabolites	Reference
1	*Escherichia coli*	4405	904	931	625	(Reed *et al.*, 2003)
2	*Saccharomyces cerevisiae*	6183	750	1149	646	(Duarte *et al.*, 2004)
3	*Haemophilus influenzae*	1775	296	488	343	(Edwards & Palsson, 1999)
4	*Helicobactor pylori*	1932	261	388	340	(Schilling *et al.*, 2002)
5	*Plasmodium falciparum*	5342	737	697	525	(Yeh *et al.*, 2004)
6	*Mannheimia succiniciproducens*	2463	335	373	352	(Hong *et al.*, 2004)
7	*Methanococcus jannaschii*	1821	436	609	510	(Tsoka *et al.*, 2004)
8	*Streptomyces coelicolor*	8042	769	700	501	(Borodina *et al.*, 2005)
9	*Staphylococcus aureus*	2702	619	641	571	(Becker & Palsson, 2005)
		--	--	773	--	(Heinemann *et al.*, 2005)
10	*Mus musculus*	~25000		1220	872	(Sheikh *et al.*, 2005)
11	*Lactococcus lactis*	2310	358	621	509	(Oliveira *et al.*, 2005)
12	*Lactobacillus plantarum*	~3110	716	645	653	(Teusink, 2005)
13	*Methanosarcina barkeri*	5072	692	619	558	(Feist *et al.*, 2006)

3.3.3 **Dual approach**

This approach uses both the reaction database and genome sequence to build the model. This approach is always superior to individual approaches, as it generates a more realistic model by filling the gaps arising in the genome centric

approach. Another advantage includes the choice of correct selection of isoenzyme for a reaction during the model reconstruction for a given organism. Table 3-1 provides the list of the published genome scale metabolic model reconstructed for various organisms. A majority of the above mentioned models were constructed using the dual approach of reconstruction of the metabolic networks, suggesting the potential of the approach in large 'omics' scale reconstruction. Ma and Zeng (2003) used this dual approach to reconstruct metabolic networks for more than 180 microorganisms using KEGG database.

3.3.4 **Automated metabolic reconstruction tools**

Almost all the databases offer a web based internet interface for browsing the content of the database. However, this feature is not suitable when we need a large amount of information to build genome scale metabolic networks from such databases. If the databases do not allow the user to download such a large amount of information selectively (e.g. a list of all the known reactions from an organism), additional tools to interact with the databases are needed. A few databases provide open access *via* web and support the use of with additional tools or libraries such as Bioperl and BioPython. Some databases have an additional SOAP protocol for accessing the information. SOAP[30] (Simple Object Access Protocol) is a tool for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts as follows:

- An envelope that defines a framework for describing what is in a message and how to process it;
- A set of encoding rules for expressing instances of application-defined data types; and
- A convention for representing remote procedure calls and responses.

SOAP libraries are also available in programming languages such as C, Python, Perl and Java.

[30] www.w3.org/TR/soap/

A few public databases are also available for complete download, and can be installed locally (e.g. KEGG, Reactome). Locally installed databases have an easy and fast access to the contents, but need to be updated periodically due to continual updating and curation over time. Another advantage of locally installed databases can be the development of in-house tools for accessing the database e.g. PyoCyc. PyoCyc was developed to access locally installed BioCyc and MetaCyc databases for the reconstruction of the metabolic networks (Poolman, PyoCyc, unpublished data).

3.4 Modelling strategy of metabolic networks

3.4.1 Model definition

After reconstruction of a metabolic network, a model is loaded for further analysis. Since the present work focuses on the structural modelling of metabolic networks, Figure 3-2 gives the details of the various steps involved in structural modelling of large metabolic networks, though the process or modelling cycle is almost the same for other modelling techniques such as kinetic modelling.

3.4.2 Model interrogation

After the reconstruction, a model needs a formal interrogation. The model interrogation process can be divided into two distinct but very important processes as described by Poolman *et al*. (2004a).

- **Constructive interrogation**

The structural metabolic model is loaded into the modelling software for constructive interrogation. This is a cyclic process involving the analysis of various aspects of structural modelling techniques for verifying the model definition. After the initial stages of the reconstruction, a systematic verification is made in order to prevent any inconsistencies in the model. For example, analysis of dead-end metabolites may suggest missing information about the utilization or production of a dead-end metabolite and to correct the model, one may need to review literature to support addition of the new reaction in the

model definition. This provides an added level of assurance for the reconstruction that the enzyme and the reaction it catalyzes do actually occur in the organism.

Francke *et al.* (2005) provide an excellent example on the need for performing the verification of the model. During a metabolic network reconstruction of *Lactobacillus plantarum* (Francke *et al.*, 2005) the model showed succinyl-CoA to be one of the reactants for a reaction that was involved in the biosynthesis of methionine. However, an understanding of the physiology of *Lactobacillus plantarum* revealed it to have an incomplete TCA cycle, and does not actually produce succinyl-CoA and uses acetyl-CoA for the biosynthesis of methionine.

Therefore, systematic verification of the initial reconstruction is needed to identify several inconsistencies that can adversely affect the final interpretation of the reconstruction, and ultimately the accurate comprehension of the molecular mechanisms of the organism.

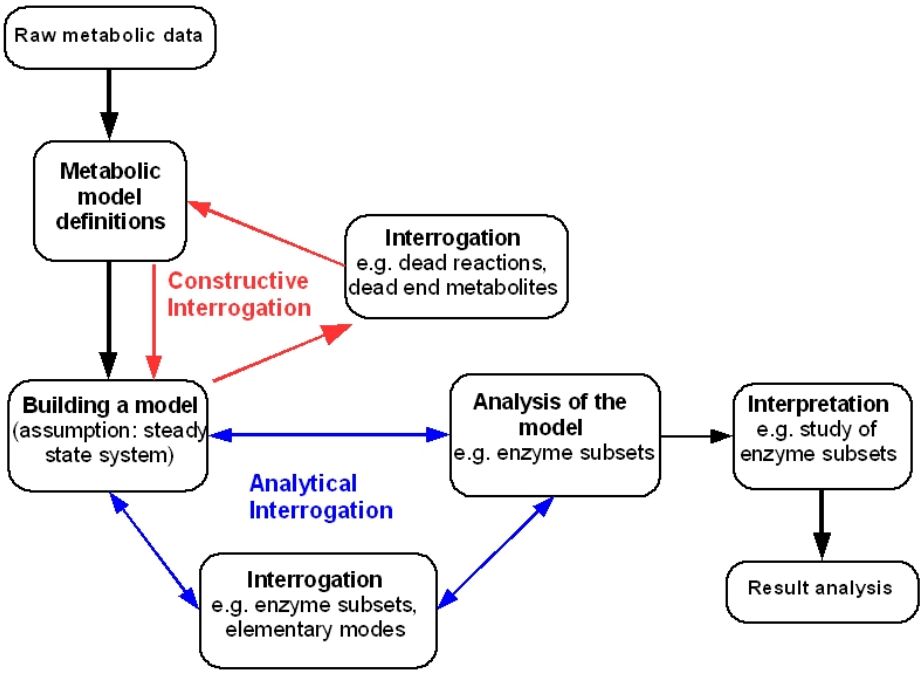


Figure 3-2 Flow chart for structural metabolic modelling

- **Analytical interrogation**

Once the model achieves the desired or satisfactory behaviour in constructive interrogation; it enters in to the next cycle called analytical interrogation, where the actual virtual experiments begin on the computer. This stage allows a number of tests where various parameters or components in a structural model, parameters such as opening or closing a transport (i.e. metabolites definition as internal/external and making reactions irreversible), can be altered to mimic the physiological conditions in the cell. Due to the *in silico* nature of the tests performed, the number of experiments that can be performed is unlimited, and experiments can be replicated many times to confirm the model behaviour.

During the analytical interrogation, the only limiting step in performing various experiments is the processing speed of computation and physical hardware status. For example, performing the elementary mode analysis for the large genome scale model (with even just 10-20 external metabolites) meets the problem of combinatorial explosion, which results in occupying the physical and then all the virtual memory of the computer.

Nevertheless, the analytical interrogation is a very important step in the modelling, as it actually produces the results for further analysis and leading to the understanding of the model behaviour and the functional capabilities of the organism of interest (Francke *et al.*, 2005).

3.4.3 **Model analysis and interpretation**

After the initial stage of interrogations, a model is ready for the interpretation of the results obtained and its further analysis. This involves the analysis of the results by various structural modelling techniques such as:

- Enzyme subsets or reaction clusters analysis;
- Elementary mode analysis (if feasible) or extreme pathway analysis;
- Damage analysis.

Analysis of enzyme subsets on a genomic scale model is discussed in Chapter 4. Elementary modes analysis is not always possible for a large genomic scale model, since its computation meets the combinatorial explosion problem.

Damage analysis can also be performed on structural metabolic models. Chapter 6 covers the details of such an analysis in the context of metabolic network modelling.

3.4.4 Advantages of model reconstruction

A reconstruction combines the relevant metabolic and genomic information of an organism, thus providing a platform to visualize and analyze 'omics' scale data. Since several inconsistencies exist between gene, enzyme, reaction databases and published literature sources on the metabolic information of an organism, reconstruction of a metabolic and genomic scale model allows a systematic verification of such diverse 'omics' data and helps in identification of 'gaps' or missing links between the genomic and metabolic data.

The reconstruction also allows metabolic comparisons between various species of the same organism as well as between different organisms. With a genome scale metabolic model, it is possible to study the topology of the metabolic network, which can be very important in understanding the metabolic capacities of the organism which subsequently aid in the metabolic engineering of an organism.

3.4.5 *E.coli* metabolic model

An *E.coli* metabolic model was reconstructed with the help of the automated model reconstruction tool 'PyoCyc' explained in Chapter 2. PyoCyc uses a local copy of MetaCyc and organism specific BioCyc databases and generates a ScrumPy model file for a given organism. EcoCyc based *E.coli* model was generated for structural analysis.

The models generated by such a process include almost all reactions from the database, resulting in a very large model of several thousand reactions. A reconstructed *E.coli* metabolic model was used to identify the challenges and problems in reconstructing and modelling of metabolic models at genomic scale. This model itself is incomplete in context to structural modelling since there are missing reaction and metabolite connections in the model. The model also

showed more than 550 reactions as dead in preliminary examination suggesting the need to curate the model definitions.

In this model, initially 1640 reactions, 1369 internal metabolites and 220 external metabolites were considered. The number of external metabolites was later increased to more than 598 due to dead-end and orphan metabolites.

3.5 Drawing metabolic networks at genomic scale

Drawing the metabolic network is always a laborious and tedious process. Since the first metabolic chart was produced (c.a. 1955) by Dr. Donald Nicholson, metabolic charts were always drawn manually in a typical graphical form (glorifying linear or cyclic pathways). The appearance of such metabolic representations in textbooks of biochemistry (Lehninger *et al.*, 1993; Stryer, 1995) made an impression of what we know as the metabolic pathways. It was obvious that their efforts were to make a simple representation of the experimental results to understand the complex biological interactions. These efforts undermined the actual complex nature of the metabolic interactions in the living cell.

While a correct definition of metabolic pathway is a matter of debate between the metabolic modelling community and biologists/experimentalists, modelling community uses mathematical approaches such as elementary modes (Schuster *et al.*, 2000), extreme pathways (Papin *et al.*, 2002) to define the metabolic pathways. Therefore, automated drawing of metabolic networks might be possible if such mathematical definitions are considered.

3.5.1 Challenges in cartography of metabolic networks

The automatic drawing of the large metabolic charts in a humanly readable form is still a tough challenge to be achieved (Michal, 1998; Schreiber, 2003). Until today, metabolic charts are drawn manually. What makes the automatic drawing of the metabolic networks so difficult when there are drawing tools available for

various other graphs or networks? Typical networks include social networks, the road network of a country etc. The study of such networks is much simpler compared to metabolic networks. A comparison between the road maps and metabolic maps shows that road maps are still simple graphs. Road maps include a node as a city or town or a junction (a joining of two or more roads) and an edge as a road. On such road maps, the location or position of the nodes and edges are generally fixed, edges have unique lengths and the angle between the two edges remain constant. While road maps do have directional constraints, the road networks are simple to understand since they are static over time. A metabolic network differs significantly from any other network. If one defines a metabolite as a 'node', and a reaction as an 'edge', one can join the substrate metabolite 'node' and product metabolite 'node' by an 'edge' (reaction). However, this representation becomes complex when a reaction has more than one substrate and product metabolites. To make things more complex, reversibility of the reaction is added to the above representation.

The problem can be partially simplified by representing each metabolite and reaction as a separate 'node' called hyper-graph and connecting them according to the directional constrains of the reaction. Such a representation is called a bipartite graph. KEGG and BioCyc (Karp *et al.*, 2005; Karp *et al.*, 2002a) uses a modified form of such bipartite graphs to represent the metabolic charts. Even this representation has some of the problems as discussed earlier. However, to some extent it is possible to automate the process of drawing a metabolic network. Such graphs differ significantly from manually drawn metabolic maps and do not serve the purpose for which they are generated, i.e. simplification of the complex structure. For e.g. Figure 3-3 shows the automated graph of *E.coli*-3 metabolic network, and appears more complex than the actual text representation of the *E.coli*-3 metabolic model.

3.5.2 **Drawing/Visualisation tools for large networks**

GraphViz[31] is an open source graph visualisation toolkit developed in the Bell Labs for visualisation and drawing of graphs and networks. Its features are as follows:

- It allows the automatic graph drawing of large networks for visualisation.
- It has several main graph layout programs such as dot, neato, twopi and circo. Dot tool can be used for generating directed network graphs.
- The GraphViz layout programs take descriptions of graphs in a simple text language, and make diagrams in several useful formats such as images and SVG for web pages, postscript for inclusion in PDF or other documents, or display in an interactive graph browser. GraphViz also supports GXL, an XML dialect.
- GraphViz has many useful features for concrete diagrams, such as options for colours, fonts, tabular node layouts, line styles, hyperlinks, and custom shapes.
- The source code for GraphViz is openly available and can be further modified. There are two Python libraries available for GraphViz which can be used for converting python data to GraphViz text layout.

The only disadvantage observed in the case of GraphViz is that when drawing large (more than few hundred nodes) metabolic network, it fails to produce output. Also like all other automatic drawing tools, the problem of placing the nodes to avoid the crossing of edges is of poor quality in GraphViz. Despite the above two shortcomings, it has a potential to become a popular tool for automatic graph layout drawing because of its open source nature.

Pajek[32] is a automated graph drawing package (Batagelj & Mrvar, 2003) that has various advantages as follows:

- It can handle very large networks with thousands of nodes and tens of thousands of links.
- Both single node and bipartite networks can be analysed.

[31] http://www.research.att.com/sw/tools/graphviz/
[32] http://vlado.fmf.uni-lj.si/pub/networks/pajek/

- It provides well published network analysis algorithms, such as network diameter (See Section 6.2.2 for definition), cluster coefficient and density determination.

Pajek uses a strictly defined, text-only file format for reading and writing networks. Although converting data sets to the Pajek .net format can be tedious as compared to the GraphViz input file, the ASCII files are humanly readable, and allow easy inspection of the raw data. The program also provides excellent network graphics, allowing networks to be displayed in a variety of layouts, once again following published algorithms. Nodes can be colour coded according to different characteristics such as component membership or degree, and nodes and links may be labelled or unlabelled. There is full control over the size, colour and position of nodes, and finished network graphics can be exported in a number of common image formats.
The major limitations of Pajek are:

- Lack of access to the underlying source code, this limits the further exploration or development of new algorithms based on tools already present in Pajek.
- The software lacks the additional scripting capability for implementation of new algorithms, platform specificity (Windows only) is another issue.
- Poor quality of documentation, and although an online manual and tutorial is provided, it is particularly difficult to find details of the algorithms implemented or the reference for the original algorithm.

3.6 Additional tools developed in Python/ScrumPy

3.6.1 Python modules developed for metabolic network interrogation

The following supporting modules were written in python for further interrogation of the metabolic network. All these modules need ScrumPy modules and objects for functioning and only work as supporting tools, though

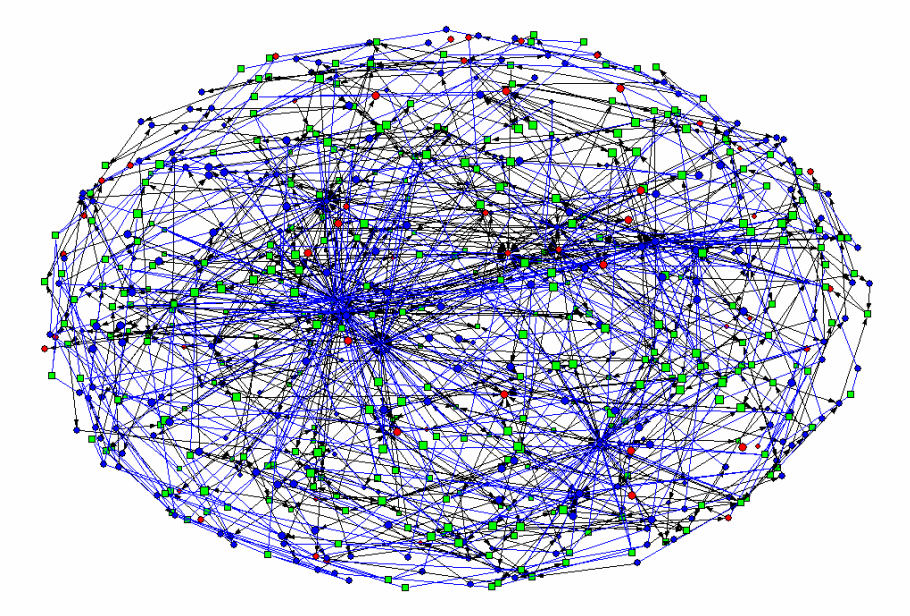


Figure 3-3 *E.coli* network generated using PyNet module of ScrumPy and Pajek network viewer.
Green squares (nodes) represent enzymes or reactions, reversible reactions are shown with blue lines (edges) while irreversible reactions have black lines (edges), red circles (nodes) represent external metabolites while blue circles (nodes) represent internal metabolites.

they are not an official part of the ScrumPy software. The python scripts for the following modules are included on the complementary CDROM media.

- PyDamage

This module was written for analysing various damage methods discussed in Chapter 6 on the ScrumPy model. More details about the definition of damage, algorithms used to design this module are discussed in Chapter 6.

- PyNet

This module converts the ScrumPy model object to graphical layout file formats. The two file formats that can be obtained from this module are GraphViz's dot file format and Pajek's network input file format. The resulting file can be used to generate the metabolic network graph of the ScrumPy model object. Figure 3-3 shows the *E.coli* model object converted from ScrumPy to Pajek network layout and visualised with Pajek's network viewer.

- Spytools

This module provides tools for connectivity analysis of stoichiometry matrix, additional tools for enzyme subset and elementary mode analysis. The module also includes scripts used for conversion of metabolic network from excel data to ScrumPy file format. It also provides tools for handling complex gene protein reaction association data.

- PyStats

Statistical analysis module used for performing Pearson and Spearman's rank correlation test, testing of multiple pair-wise correlations using Bonferroni-Holm (BH) and Sidak-Holm (SH) test.

3.7 Discussion

To summarise, the automated reconstruction of large 'omics' scale metabolic model is possible using software tools such as PyoCyc. Such models still need a high level of manual intervention during reconstruction process. Yet, such models provide a better option as a starting point for reconstruction of metabolic models, compared to laborious manual addition of individual reactions.

There is a need to tackle some challenges in the databases to accelerate the efforts of automatic reconstruction for various other organisms. The need is not just to improve the quality and quantity of the data in a database but also to develop the universal data representation format for the biological data (Poolman *et al.*, 2006).

Even though there are no set guidelines for the validation process for structural metabolic model, the model should represent the snap shot of an organism's metabolism as correctly as possible. Structural modelling techniques discussed in the Chapter 2 can provide aid in the modelling and validation of a large 'omics' scale metabolic model. Such techniques can be successfully applied to understand the metabolic capacities of the system and give a flexible way to perform various *in silico* experimental techniques on the model.

Chapter 4
Understanding the substructure of large metabolic networks

4.1 Modelling of *E.coli* structural networks

Due to the availability of large metabolic, genomic and proteomic data, large genomic scale structural metabolic models of *E.coli* were obtained from various published data. Three models were used to represent moderate size, large size and very large (database) size genomic scale model. Table 4-1 gives the details of model specifications and comparison on the size of the model.

Table 4-1 Specification of the *E.coli* models used for the present study

	Model name	Micro-organism	Model specification			Reference
			Reactions	Metabolites		
				Internal	Externals	
1	*E. coli-1*	*E. coli*	338	260	51	(Fell & Wagner, 2000)
2	*E. coli-2*	*E. coli*	730	582	52	(Edwards *et al.*, 2001)
3	*E. coli-3*	*E. coli*	935	762	68	(Reed *et al.*, 2003)
4	*E. coli-4*	*E. coli*	1640	1369	220	Bonde, this dissertation, unpublished data[33]

4.1.1 *E.coli*-1 model specifications

This model was taken from (Wagner & Fell, 2001) and was based on the manually curated reactions of *E.coli* (Pramanik & Keasling, 1997). The model only includes reactions involving small chemical metabolites and covering the few essential metabolic systems in a cell, such as carbohydrate metabolism.

This model includes 338 reactions, 260 metabolites and 51 external metabolites. It includes reactions from glycolysis (12 reactions)[34], pentose phosphate and Entner-Doudoroff pathways (10), glycogen metabolism (5), acetate production (2), glyoxalate and anaplerotic reactions (3), tricarboxylic acid cycle (10), oxidative phosphorylation (6), amino acid and polyamine biosynthesis (95), nucleotide and nucleoside biosynthesis (72), folate synthesis and 1-carbon

[33] EcoCyc based model generated using PyoCyc, reconstruction is discussed in Chapter 3
[34] Number in brackets indicates the number of reactions present in the metabolic subsystem.

metabolism (16), glycerol 3-phosphate and membrane lipids (17), riboflavin (9), coenzyme A (11), NAD(P) (7), porphyrins, haem and sirohaem (14), lipopolysaccharides and murein (14), pyrophosphate metabolism (1), transport reactions (2), glycerol 3-phosphate production (2), isoprenoid biosynthesis and quinine biosynthesis (13). Initially, this model was used for interrogation because of its moderate size.

4.1.2 *E.coli*-2 and *E.coli*-3 model specifications

These two models were taken from Prof. Palsson's online model repository[35]. *E.coli*-2 used 730 reactions and 582 metabolites (Edwards et al., 2001). This model was later extended to *E.coli*-3 model, which included 905 reactions, 762 metabolites, and 68 external metabolites (Reed *et al.*, 2003). *E.coli*-3 model was extensively studied in detail using various structural modelling techniques.

The *E.coli*-3 model covers the following metabolic reactions: Carbon metabolism (TCA cycle (13), glycolysis (18), pentose phosphate cycle (10) alternate carbon metabolism (130)), anaplerotic (7) reactions, cell envelop synthesis (80), cofactor and prosthetic group biosynthesis (135), membrane lipid (35), nucleotide salvage (86), purine and pyrimidine biosynthesis (24 reactions), oxidative phosphorylation (40), extra cellular transport (184), amino acid metabolism (valine, leucine, and isoleucine (15), tyrosine, tryptophan, and phenylalanine (20), threonine and lysine (14), histidine (10), glycine and serine (8), cysteine (8), glutamate(10), arginine and proline (43), alanine and aspartate (10)] and 28 other reactions which include nitrogen, oxygen metabolism and putative reactions. These reactions were associated with 876 genes on the gene-protein-reaction associations.

The gene-protein-reaction associations produced by Reed *et al.* (2003) were further investigated and a few inconsistencies were corrected *via* addition of transport and missing reactions. An updated list of gene-protein-reaction association for *E.coli* can be found on the complementary CDROM media.

[35] http://gcrg.ucsd.edu/organisms/ecoli.html

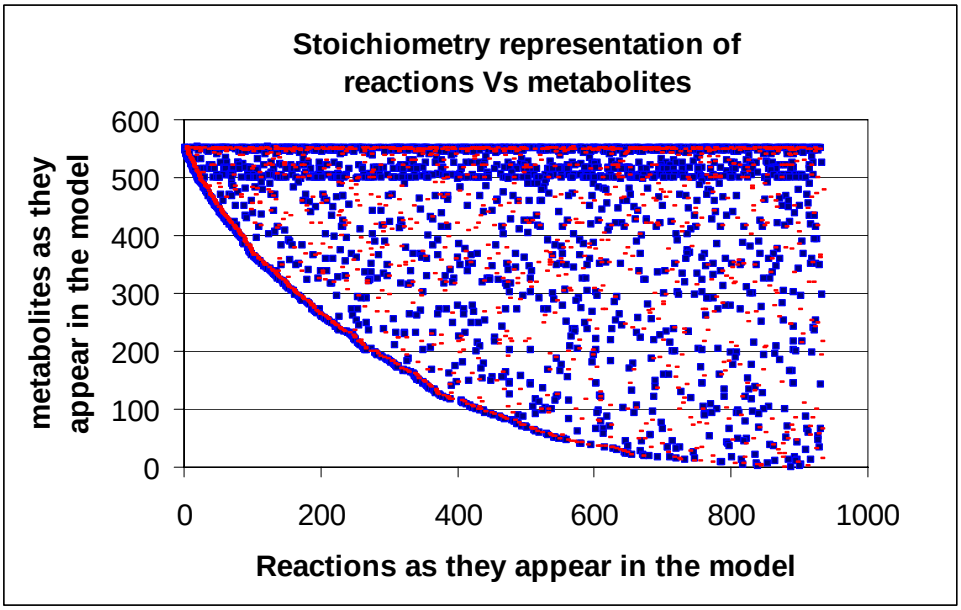


Figure 4-1 The dot plot representation of the stoichiometry matrix of *E. coli*-3 model (A blue dot indicates the positive element while a red dash indicates the negative element in the stoichiometry matrix).

Figure 4-1 shows a stoichiometry matrix of the model. The x-axis shows the position of a reaction and the y-axis shows the position of a metabolite in the stoichiometry matrix. A blue dot represents positive stoichiometric coefficient (>0), while a red dash (hyphen) indicates a negative stoichiometric coefficient (<0) in the stoichiometry matrix. The graph shows that a few metabolites are involved in high number of reactions while most of the metabolites are only involved in very few reactions in the model.

Few manual corrections were made in the E.coli-3 model after the constructive interrogation. Table 4-2 gives the list of original external metabolites defined by Reed *et al.* (2003). After a connectivity check on the model, metabolites shown in Table 4-3 appeared only once in the model, suggesting that these metabolites are orphan metabolites. All such metabolites were defined as externals in the model.

Table 4-2 List of external metabolites in the original *Ecoli*-3 model defined by Reed *et al.*(2003)

No.	External metabolite	No.	External metabolite
1	1,5-Diaminopentane_EXTCELL	72	Lactose_EXTCELL
2	2-Dehydro-3-deoxy-D-gluconate_EXTCELL	73	L-Alanine_EXTCELL
3	2-Oxoglutarate_EXTCELL	74	L-Arabinose_EXTCELL
4	3-(3-hydroxy-	75	L-Arginine_EXTCELL

	phenyl)propionate_EXTCELL		
5	3-hydroxycinnamic acid_EXTCELL	76	L-Asparagine_EXTCELL
6	4-Aminobutanoate_EXTCELL	77	L-Aspartate_EXTCELL
7	Acetaldehyde_EXTCELL	78	L-Carnitine_EXTCELL
8	Acetate_EXTCELL	79	L-Cysteine_EXTCELL
9	Acetoacetate_EXTCELL	80	L-Fucose 1-phosphate_EXTCELL
10	Adenine_EXTCELL	81	L-Fucose_EXTCELL
11	Adenosine_EXTCELL	82	L-Glutamate_EXTCELL
12	Allantoin_EXTCELL	83	L-Glutamine_EXTCELL
13	Ammonia_EXTCELL	84	L-Histidine_EXTCELL
14	Butyrate (n-C4:0)_EXTCELL	85	L-Idonate_EXTCELL
15	Choline_EXTCELL	86	L-Isoleucine_EXTCELL
16	Citrate_EXTCELL	87	L-Lactate_EXTCELL
17	CO2_EXTCELL	88	L-Leucine_EXTCELL
18	Cob(I)alamin_EXTCELL	89	L-Lysine_EXTCELL
19	Cyanate_EXTCELL	90	L-Malate_EXTCELL
20	Cytidine_EXTCELL	91	L-Methionine_EXTCELL
21	Cytosine_EXTCELL	92	L-Phenylalanine_EXTCELL
22	D-Alanine_EXTCELL	93	L-Proline_EXTCELL
23	Deoxyadenosine_EXTCELL	94	L-Rhamnose_EXTCELL
24	Deoxycytidine_EXTCELL	95	L-Serine_EXTCELL
25	Deoxyguanosine_EXTCELL	96	L-tartrate_EXTCELL
26	Deoxyinosine_EXTCELL	97	L-Threonine_EXTCELL
27	Deoxyuridine_EXTCELL	98	L-Tryptophan_EXTCELL
28	D-Fructose_EXTCELL	99	L-Tyrosine_EXTCELL
29	D-Galactarate_EXTCELL	100	L-Valine_EXTCELL
30	D-Galactonate_EXTCELL	101	Maltohexaose_EXTCELL
31	D-Galactose_EXTCELL	102	Maltopentaose_EXTCELL
32	D-Galacturonate_EXTCELL	103	Maltose_EXTCELL
33	D-Glucarate_EXTCELL	104	Maltotetraose_EXTCELL
34	D-Gluconate_EXTCELL	105	Maltotriose_EXTCELL
35	D-Glucosamine_EXTCELL	106	Melibiose_EXTCELL
36	D-Glucose 6-phosphate_EXTCELL	107	meso-2,6-Diaminoheptanedioate_EXTCELL
37	D-Glucose_EXTCELL	108	N-Acetyl-D-glucosamine_EXTCELL
38	D-Mannose 6-phosphate_EXTCELL	109	N-Acetyl-D-mannosamine_EXTCELL
39	D-Glyceraldehyde_EXTCELL	110	N-Acetylneuraminate_EXTCELL
40	Dihydroxyacetone_EXTCELL	111	Nicotinate_EXTCELL
41	Dimethyl sulfide_EXTCELL	112	Nitrate_EXTCELL
42	Dimethyl sulfoxide_EXTCELL	113	Nitrite_EXTCELL
43	D-Lactate_EXTCELL	114	NMN_EXTCELL
44	D-Mannitol_EXTCELL	115	O2_EXTCELL
45	D-Glucuronate_EXTCELL	116	octadecanoate (n-C18:0)_EXTCELL
46	D-Mannose_EXTCELL	117	Ornithine_EXTCELL
47	D-Methionine_EXTCELL	118	Phenylpropanoate_EXTCELL
48	D-Ribose_EXTCELL	119	Phosphate_EXTCELL
49	D-Serine_EXTCELL	120	Putrescine_EXTCELL
50	D-Sorbitol_EXTCELL	121	Pyruvate_EXTCELL
51	D-Xylose_EXTCELL	122	R-Pantothenate_EXTCELL
52	Ethanol_EXTCELL	123	Sodium_EXTCELL

53	Fe2_EXTCELL	124	Spermidine_EXTCELL
54	Formate_EXTCELL	125	S-Propane-1,2-diol_EXTCELL
55	Fumarate_EXTCELL	126	Succinate_EXTCELL
56	Galactitol_EXTCELL	127	Sucrose_EXTCELL
57	gamma-butyrobetaine_EXTCELL	128	Sulfate_EXTCELL
58	Glycerol 3-phosphate_EXTCELL	129	Taurine_EXTCELL
59	Glycerol_EXTCELL	130	tetradecanoate (n-C14:0)_EXTCELL
60	Glycine betaine_EXTCELL	131	Thiamin_EXTCELL
61	Glycine_EXTCELL	132	Thiosulfate_EXTCELL
62	Glycolate_EXTCELL	133	Thymidine_EXTCELL
63	Guanine_EXTCELL	134	Trehalose_EXTCELL
64	Guanosine_EXTCELL	135	Trimethylamine N-oxide_EXTCELL
65	H_EXTCELL	136	Trimethylamine_EXTCELL
66	H2O_EXTCELL	137	Uracil_EXTCELL
67	Hexadecanoate (n-C16:0)_EXTCELL	138	Urea_EXTCELL
68	Hypoxanthine_EXTCELL	139	Uridine_EXTCELL
69	Indole_EXTCELL	140	Xanthine_EXTCELL
70	Inosine_EXTCELL	141	Xanthosine_EXTCELL
71	K_EXTCELL		

Table 4-3 List of metabolites made external in the E.coli-3 model after connectivity analysis or constructive interrogation[36].

No.	External metabolite	No.	External metabolite
1	Nicotinamide adenine dinucleotide – reduced	34	E-3-carboxy-2-pentenedioate 6-methyl ester
2	Nicotinamide adenine dinucleotide	35	5,6-Dimethylbenzimidazole
3	Arbutin 6-phosphate	36	Heme O
4	Hydroquinone	37	Siroheme
5	2-Phosphoglycolate	38	Enterochelin
6	Nicotinamide adenine dinucleotide phosphate	39	S-Adenosyl-4-methylthio-2-oxobutanoate
7	Nicotinamide adenine dinucleotide phosphate – reduced	40	4-Amino-5-hydroxymethyl-2-methylpyrimidine
8	trans-Cinnamate	41	Chorismate
9	Phenylacetyl-CoA	42	IDP
10	3-keto-L-gulonate-6-phosphate	43	4-Hydroxy-benzyl alcohol
11	Aminoacetaldehyde	44	4-Hydroxy-L-threonine
12	Phenethylamine	45	1-deoxy-D-xylulose
13	2,3-Dioxo-L-gulonate	46	glycogen
14	3-Dehydro-L-gulonate	47	Cardiolipin
15	N1-Acetylspermidine	48	4-hydroxy-5-methyl-3(2H)-furanone
16	N8-Acetylspermidine	49	ITP
17	Peptidoglycan subunit of Escherichia coli	50	4-Methyl-5-(2-hydroxyethyl)-thiazole
18	Enterobacterial common antigen polysaccharide	51	P1,P4-Bis(5^-adenosyl) tetraphosphate
19	P1,P5-Bis(5^-adenosyl) pentaphosphate	52	P1,P4-Bis(5^-guanosyl) tetraphosphate

[36] This table include both, currency metabolites and orphan metabolites present in the model.

20	UDP-D-galacto-1,4-furanose	53	GDP-L-fucose
21	Glycerophosphoserine	54	cAMP
22	Sn-Glycero-3-phospho-1-inositol	55	Thymine
23	KDO(2)-lipid IV(A)	56	crotonobetaine
24	cold adapted KDO(2)-lipid (A)	57	Maltose 6^-phosphate
25	Phosphatidylcholine	58	D-Methionine
26	acyl phosphatidylglycerol	59	L-Fucose 1-phosphate
27	lipopolysaccharide	60	Cyanide
28	dTDP-L-rhamnose	61	Thiocyanate
29	1D-myo-Inositol 1-phosphate	62	Selenide
30	UDP-D-glucuronate	63	Selenophosphate
31	apoprotein [acyl carrier protein]	64	Superoxide anion
32	Oxidized glutathione	65	trans-Aconitate
33	Pimeloyl-CoA	66	Cobinamide

4.1.3 **Gene-protein-reaction (GPR) association for *E.coli***

The gene-protein-reaction (GPR) associations are much more complex than it appears when the genome, proteome and reactome of an organism are considered together. This major problem of gene protein reaction association complexity is due to the highly complex nature of the associations between genes, proteins and reactions. The association between the gene and protein to reaction is not one to one, but is more complex. As shown in Figure 4-2, the trivial case is one gene to one protein to one reaction. However, it is also possible that two or more genes gives one protein which then catalyses two or more reactions (shown with genes b0071 and b0072 in Figure 4-2). The complexity increases when any one of the more than two genes gives the same protein. One protein can catalyse two different reactions or two different proteins catalyse different reactions.

Gene protein reaction associations were obtained from Reed *et al*. (2003). The associations were not a simple one-gene-to-one-protein-to-one reaction, but rather quite complex. In the *E. coli* genome, few enzymes catalyze multiple reactions; few reactions are catalyzed by multiple enzymes. This scenario is even more complex, when enzyme (or protein) to gene associations are considered. One gene may express one protein, or two or more genes may express different domains of the same or different proteins. Proteins (subunits) combine together to form an active form of an enzyme which catalyses one or more reactions. In addition, in *E.coli* there are a few reactions that are known to

be catalyzed by enzymes, wherein the enzyme itself has yet to be identified, or the gene encoding the protein is yet to be identified. This makes the GPR associations complex and incomplete.

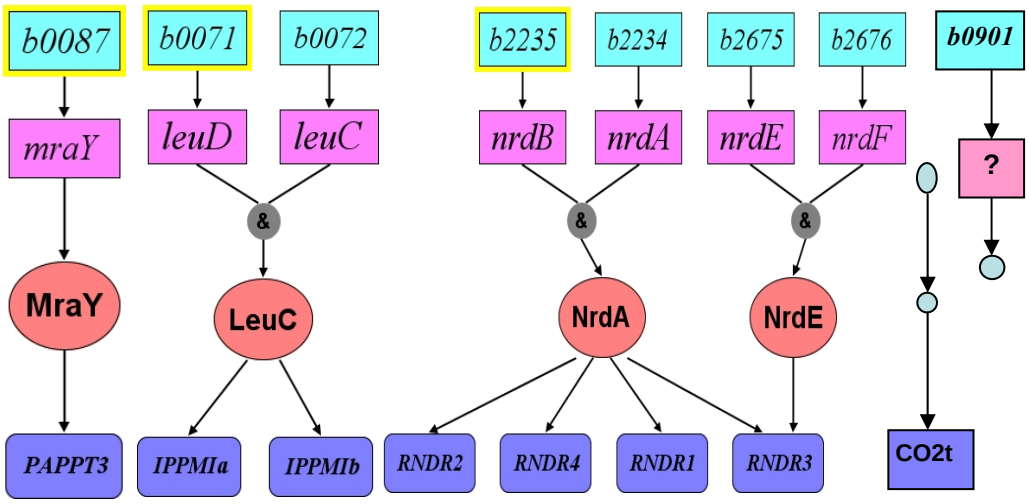


Figure 4-2 Complex Gene-Protein-Reaction (GPR) assignment in *E.coli*

A refined GPR association table for *E .coli* is included in the supporting material on the CDROM.

4.1.4 Introduction to operon and regulon

The classical definition of the operon as given by Jacob and Monod (1961) is 'a group of two or more genes transcribed as a polycistronic unit'. With the discovery of lactose operon, they suggested that *E.coli* does not waste energy in making enzymes for lactose metabolism, when a simpler sugar, such as glucose, is available and suggested the importance of the feedback in biological systems. The 'operon model' proposed by Jacob and Monod (1961) later became a paradigm for understanding the gene regulation in all organisms and led to the study of modelling genetic regulatory mechanisms in bacteria using feedback systems.

The present definition of operon is 'a set of adjacent structural genes in bacteria whose mRNA is synthesized in one piece, together with the adjacent regulatory signals that affect transcription of the structural genes' (Salgado *et al.*, 2004). Such a sequence of adjacent bacterial genes functions together under the

transcriptional control of the same operator with an all-or-none response[37]. Identification of such operons is important in understanding the regulation of the metabolic systems. The importance of the study of operons is well reported in literature (Ermolaeva *et al.*, 2001; Itoh *et al.*, 1999; Tjaden *et al.*, 2002a).

Operons facilitate efficient implementation of transcriptional regulation in microbial genome. They also provide useful information for the characterization and reconstruction of biological and metabolic networks, as genes in an operon tend to have physiologically related functions.

Another similar discovery in the same decade was the 'regulon'. The initial definition given by Maas (1964) is a 'set of genes subject to regulation of one and only one regulator'. This definition was derived from studies of the arginine biosynthesis genes, which were, contrary to operons, found to be scattered (non-contiguously located) on the chromosome in *E.coli*. Later such clusters were also observed in eukaryotes, giving a modified definition as a genetic unit consisting of a non-contiguous group of genes under the control of a single or many regulator genes. In bacteria, regulons are global regulatory systems and may involve in the interplay of pleiotropic[38] regulatory domains consisting of several operons (Salgado *et al.*, 2004). With the development of microarray technology for whole genome expression profiling, the identification of regulons is now possible for various organisms (Ogura *et al.*, 2001). The ODB database further classifies the operons as

- **Known operon**

In such an operon, the region transcribed can be confirmed by experiments such as Northern hybridization. If any experimental documentation is present or reported as transcription unit in literature, the operon is called known operon.

- **Putative operon**

Putative operons are defined by orthologous genes and their location on the genome. If all the orthologous genes are observed in the genome, or if the orthologous genes are consecutively located on the genome, then this group of

[37] 'All or none' response is just a matter of definition – all the genes certainly, but not either completely 'on' or completely 'off'.
[38] Producing many or multiple effects from a single gene.

genes can be assigned as putative operon for a given species. One can also identify such a putative operon if all the genes are in a cluster of orthologous genes, and form a known operon in other organism.

Table 4-4 Operon classified on functional/metabolic system in *E.coli* (Daruvar *et al.*, 2002)

Metabolic pathway /subsystem	Number of operons	Total operons
1. Energy metabolism		64
1.1. Biosynthesis of cofactors	8	
1.2. Amino acid biosynthesis	11	
1.3. Carbon compound catabolism	9	
1.4. Central intermediary metabolism	9	
1.5. Energy metabolism	24	
1.6. Fatty acid and phospholipid ...	1	
1.7. Nucleotide biosynthesis ...	1	
1.8. Putative enzymes	1	
2. Information		13
2.1. DNA replication, recombination	3	
2.2. Transcription, RNA processing	1	
2.3. Translation, posttranslation	9	
3. Regulation		1
3.1. Regulatory function		
3.2. Putative regulatory proteins	1	
4. Transport		32
4.1. Putative transport proteins	1	
4.2. Transport and binding proteins	31	

The *E.coli* operon data was taken form two different databases; RegulonDB (Salgado *et al.*, 2004) which reports 852 operons for *E.coli* K12 MG 1655 and Operon Database (ODB) (Okuda *et al.*, 2006) which reports 823 operons for *E.coli* K12.

RegulonDB operons were further classified on the basis of genes involved in metabolic subsystem as suggested by Daruvar *et al.* (2002). Table 4-4 shows the operon classification on the basis of genes associated to *E.coli*-3 model in the present study. Operon is associated to the metabolic subsystem if one or more operonic gene encodes reactions in that subsystem in *E.coli*-3 model.

4.2 Analysis of *E.coli* models

First, the *E. coli*-1 model was used to identify the challenges in analysing large models. The *E. coli*-1 model was tested for computation of enzyme subsets and elementary modes on the Athlon AMD processor with 512 MB RAM. Later, the

E. coli-3 model was taken for analysis, after a manual curation for the blocked reactions, dead-end metabolites, orphan metabolites and gene-protein-reaction associations.

4.2.1 **Orphan and dead-end metabolites analysis**

All the models were constructively interrogated for dead-end and orphan metabolites. Orphan and dead-end metabolites were determined by examination of the stoichiometry matrix of the models. Rows of ***N*** with only one non-zero element correspond to orphan metabolites, while rows whose non-zero elements all have the same algebraic sign ('+' or '-') and reactions in which such metabolites involved are all irreversible, correspond to dead-end metabolites. However, the identification of dead-ends is not always assured from the above suggested approach. A few interesting cases were observed in *E.coli*-3 model where metabolites that did not satisfy the criteria for dead-ends, were observed to be dead ends. One such case is shown in Figure 4-3 where external thiamine is transported *via* THMabc (thiamine ABC transporter) reaction. This reaction is an alternative reaction to thiamine biosynthesis in *E.coli*. The thiamine biosynthesis in *E. coli* involves production of thiamine monophosphate from hydroxymethylpyrimidine.

As shown on the Figure 4-3, EC-2.7.4.7-PMPK and EC-2.5.1.3 are irreversible reactions, producing thiamine monophosphate. The EC-2.7.1.89-TMKr reaction converts thiamine to thiamine monophosphate. However, this reaction is reversible, resulting in either side of the metabolite as dead-end. There is only one reaction for transport of Thiamine (from ThiamineExt) and only one reaction producing Thiamine Monophosphate. In the stoichiometry matrix, such a metabolite row includes two non-zero elements with two different algebraic signs respectively (negative element for reaction consuming and positive element for reaction producing this metabolite), therefore, will not be identified by simple dead-end search algorithm based on stoichiometric analysis.

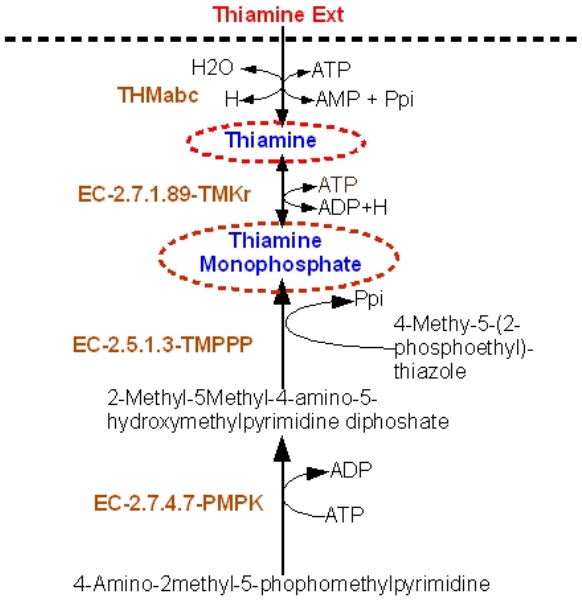


Figure 4-3 A case study of dead-end analysis from *E.coli*-3 model

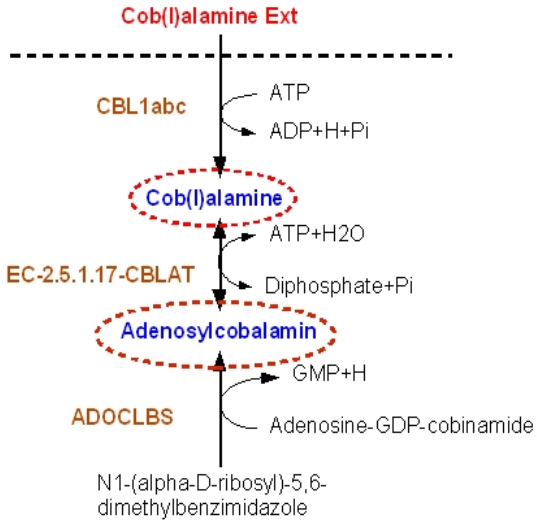


Figure 4-4 Dead-end metabolite identification

A second case is explained in Figure 4-4. The cob(I)alamin uptake *via* ABC transport is an alternative resource of vitamin B_{12} in *E.coli* and plays an important role in reactions such as cobalamin-dependent homocysteine transmethylase. However, the lack of further utilisation of cob(I)alamin and adenosylcobalamin results in dead-end formation in the metabolic network. Thus this identification of dead-ends provides hints about the missing links or

information on reaction reversibility in model definition and aids in validation of the model.

4.2.2 **Dead enzyme or reaction analysis**

While structural modelling is based on the pseudo steady state assumption, such models show the presence of a few reactions that do not carry any active flux. Such reactions, as discussed in section 2.3.3, point out incorrect model definition; identification and analysis of such reactions may help in model validation. Such dead reactions are placed into a common set during the enzyme subsets computation and are called 'dead enzyme subset' in ScrumPy.

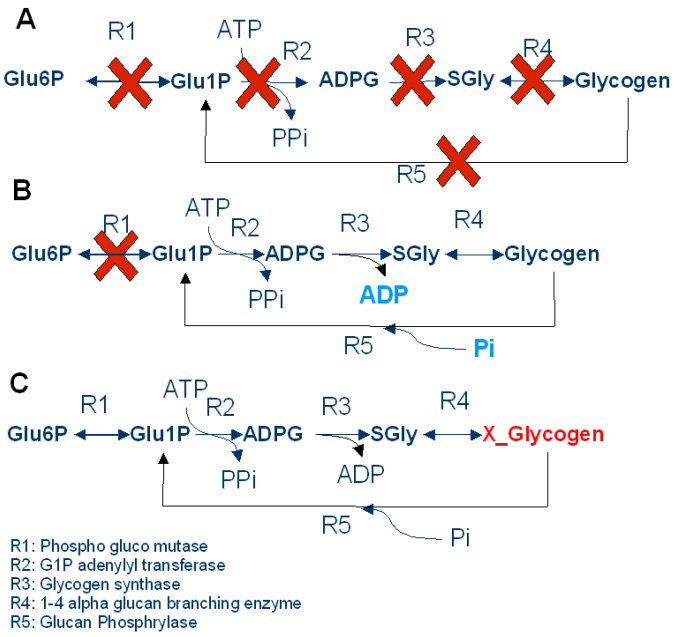


Figure 4-5 Dead reaction analysis.
A) A side branch of 5 reactions was found dead in the *E.coli*-1 network. B) Analysis of all 5 reactions revealed that there are missing metabolites in reaction R2 and R5 which were stoichiometrically corrected. C) The reaction R1 was dead because of futile cycle formation of R2, R3, R4 and R5; defining Glycogen as external allows flux through the reaction.

One such example is shown in Figure 4-5. During the analysis of the *E.coli*-1 model, a set of 5 reactions (labelled as R1 to R5 in Figure 4-5) including a phosphoglucomutase (R1), was found dead in the model. A further analysis of all the reactions was performed to find the reasons for these reactions being dead. Two errors were found in reaction R3 and R5. R3 and R5 were stoichiometrically unbalanced with one metabolite missing in each reaction;

missing metabolites were added to correct the reaction stoichiometry. However, after correction of the two errors, a new analysis of dead reactions showed reaction R1 still being dead. The problem observed in this case was due to the occurrence of a futile cycle in the system (explained in next subsection). The cyclic flux through reactions R2, R3, R5 and R5 leads to no flux through R1, as the steady state criteria are satisfied for both, substrate and product of the R1. To mimic the physiology of glycogen synthesis, glycogen storage is assumed by defining glycogen as an 'external' metabolite. R1 then becomes active, since additional flux now passes through R1, R2, R3 and R4, resulting into the production of glycogen.

4.2.3 **Substrate cycles identification**

Identification of substrate or futile cycles needs computation of elementary modes on a model. Elementary modes with 'zero' net production and consumption are 'futile modes'. Since computation of elementary modes on genomic scale large metabolic models is computationally challenging, a modified subroutine was implemented to obtain the futile cycles.

In the modified subroutine, all the transport reactions (i.e. reactions that produce or utilise external metabolite) and all the external metabolites were removed from the model. With an assumption that the model is a closed network (no source and sink metabolites and reactions), this modified model was used to identify dead reactions using the null space approach. The dead reactions were eliminated from the model (i.e. from stoichiometry matrix) to produce another with a reduced number of reactions, and then elementary mode analysis was performed. The resulting computation returns only the futile cycle modes since the external metabolites were eliminated from the network.

Pseudo code for futile (internal) cycle computation
sm= stoichiometry matrix

i) Get all transport reactions
ii) Delete all the transport reactions from sm
iii) Remove metabolite (rows) with all zero elements of sm, if any
iv) Compute the null space (**K**) for sm and identify all dead reactions
v) Remove all dead reactions from sm space.
vi) Convert the new sm to new ScrumPy model

viii) Load new ScrumPy model and compute elementary model, all elementary modes are futile cycles in the model.

This modified algorithm efficiently produces all the internal cycles for large genomic scale models. Table 4-5 shows the results on internal cycles obtained for *E.coli* models.

Table 4-5 Results for Futile cycles for *E.coli* models

Model	Model Size (Reactions)	Number of Futile cycles
E.coli-1	338	504
E.coli-3	935	182

4.3 Enzyme subset study

The main objective of the present study was to identify groups of enzymes which operate at the same time in a fixed proportion. The study of enzyme subsets computed for various *E.coli* models is discussed in this section.

4.3.1 *E.coli*-1 model enzyme subset analysis

In the *E.coli*-1 model, the subset analysis shows that more than 75 percent of the reactions belong to enzyme subsets of size two or more reactions (Figure 4-6). Almost 77 reactions appear as a single reaction subset and additional 14 reactions in the dead (no flux carrying reaction) subset. Because of the small number of large reaction clusters, the large subsets were individually studied.

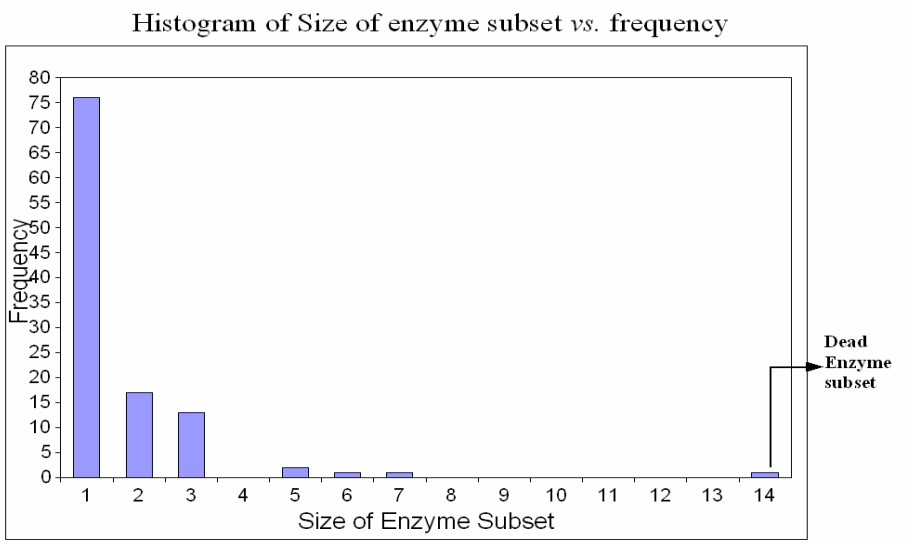


Figure 4-6 Histogram of size of enzyme subsets vs. frequency of subsets in *E.coli*-1

4.3.2 *E.coli*-3 model enzyme subset analysis

The *E.coli*-3 model was analysed for the enzyme subsets. Figure 4-7 shows the initial analysis of the model which resulted in a dead subset of 70 reactions. Further analysis of dead reactions helped in model curation.

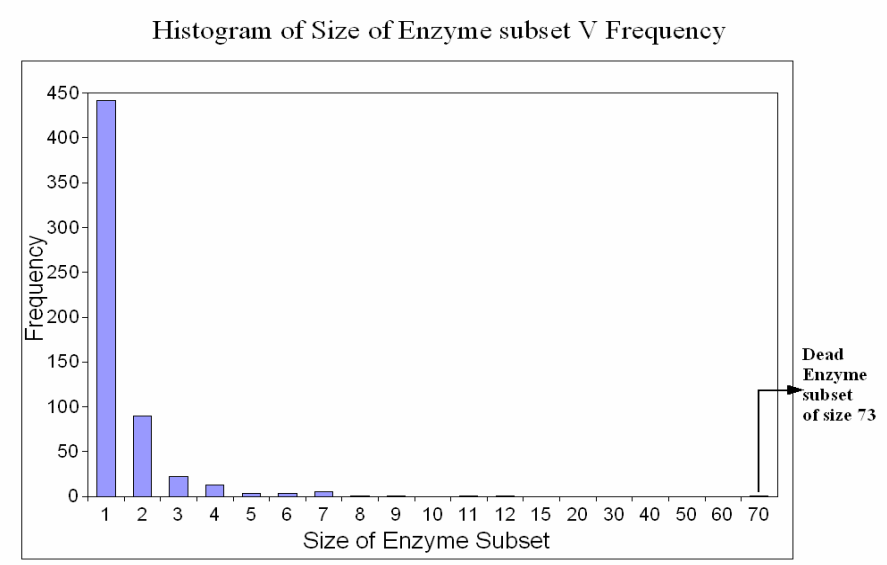


Figure 4-7 Histogram of size of enzyme subset *vs*. frequency of subsets for *E.coli*-3 model before constructive model interrogation

After the constructive interrogation step, the final model showed a total of 622 subsets including one dead subset of 40 enzymes. The model shows that 50 percent of the reactions always occur in enzyme subsets with varying sizes of two or more reactions. Figure 4-8 shows the size distribution of the enzyme subsets in the curated *E.coli*-3 model.

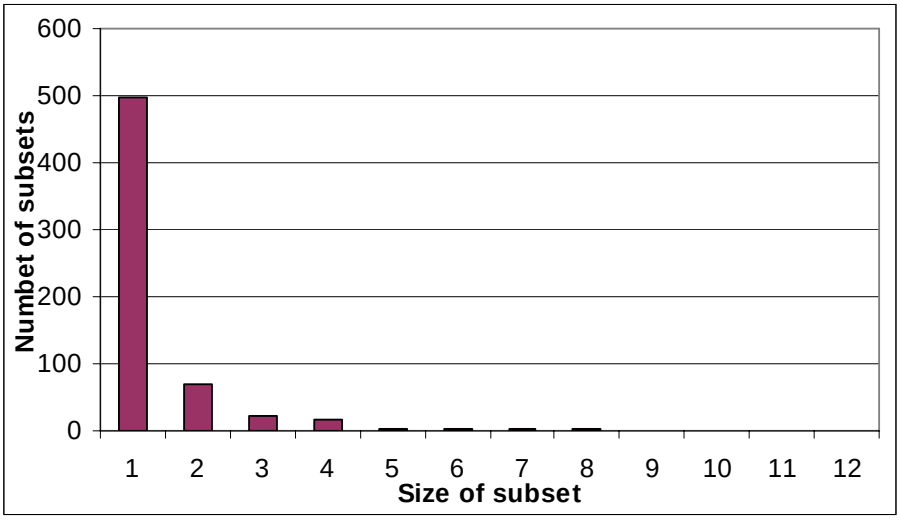


Figure 4-8 Enzyme subset vs. frequency of subsets for *E.coli*-3 model

(After constructive interrogation and correcting model definition, size of enzyme subset vs. frequency plot for *E.coli*-3 model shows increase in number of subsets of sizes 1 and 2)

4.4 Scope of study

Figure 4-9 gives the overview of the present study. For a large 'omics' scale metabolic network, reactions in a subset always carry flux in a fixed proportion. It is possible that such a group of reactions with 'all or none' flux carrying relationship might be sharing some common regulatory structure on the genome. By analogy, there must be a similar cluster or group of genes on the genome of the organism which might control the expression of the genes responsible for encoding reactions in such cluster. On the genome, it is well known that an operon is a group of adjacent genes which share a common regulatory pattern.

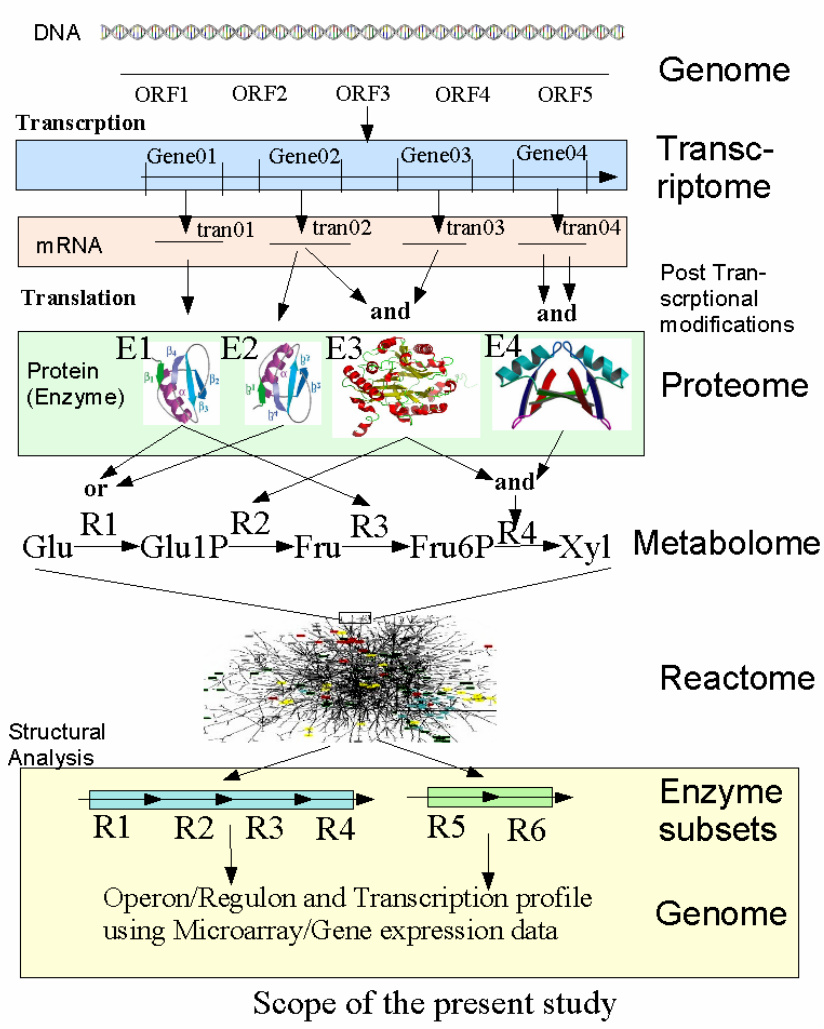


Figure 4-9 Overview of the genomic scale structural modelling

4.4.1 **Operons and enzyme subsets**

To check the possibility of some correlation between the metabolic reaction sets and genome structure, all the reactions in a subset were associated with the genes responsible for carrying the reactions. Some of the genes cluster as operons in *E.coli* and genes in an operon share a common regulatory protein and such genes are usually functionally related to each other.

The *E.coli*-1 model was first chosen to find the correlation between its subsets and operons. Due to its moderate size, a manual association of gene to reactions in a subset was carried out to find a possible similarity with operons on *E.coli* genome. During the analysis, as shown in Figure 4-10, a subset of three glycogen synthesis reactions and genes responsible for these reactions were found in an operon on the *E.coli* genome.

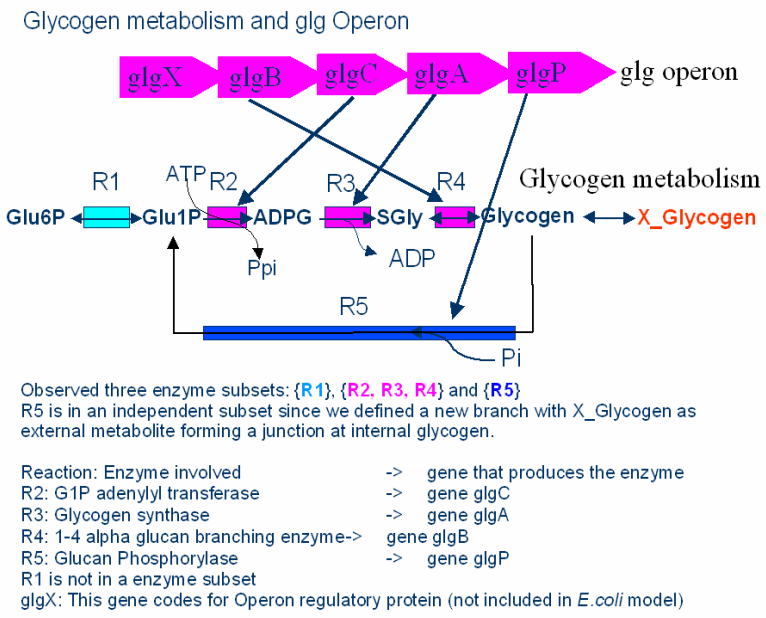


Figure 4-10 An example of enzyme subset to operon correlation in *E.coli*-1 model

Table 4-6 Subset to operon correlation for *E.coli*-1 model

Model	Total subsets	Matching criteria	Matching Criteria (Exact/partial)	Number of matches
E.coli-1	263	subset to operon	Exact (Class-1)	31
			Partial (Class-2)	61

As shown in Figure 4-10, the glg operon consists of five genes, three of the genes (glgB, glgC and glgA) are directly responsible for the reactions in a subset which synthesize glycogen, while glgX codes for operon regulatory protein and is not considered in *E.coli*-1 metabolic model. The only gene glgP, which codes for the reaction which is not in a subset of glycogen synthesis, might also have appeared in a subset if glycogen was not declared as external (discussion in Section 4.2.3 on dead reactions). To identify further similar subset-to-operon matches, all the other enzyme subsets from *E.coli*-1were analysed manually. The *E.coli*-1 model consists of 263 subsets. Of these, 78 subsets contain only one reaction. The subset-to-operon correlation was performed only for those subsets that contain two or more reactions.

Even for this moderate size model (*E.coli-1*), more than 10 percent of subsets show a perfect match with the operon data on the genome. The number of

partial matches with operons is around 20 percent of the total subsets as shown in Table 4-6. Nevertheless, this data suggested that there might be some correlation between the enzyme subsets and operon structure, leading to the analysis of the similar correlations in the *E.coli*-3 model.

The classification of operons based on the functional or metabolic systems in *E.coli* as shown earlier in Table 4-4 was adopted in the present study (Daruvar *et al.*, 2002). The *E.coli*-1 model accounts for almost all the energy metabolism reactions, therefore covers all the operon genes in class 1 of Table 4-4. When compared with Daruvar *et al.*(2002) classification of operon, more than 50 percent of the *E.coli* operons from carbon metabolism class show similar structural matching with the enzyme subsets in *E.coli*-1 model.

4.4.2 Comparison between RegulonDB and ODB database

To check the quality of the operon data from two databases, a comparative study on the presence of an operon in the two databases was made. A significant difference was observed in the two databases. While 585 operons were found to match exactly in terms of the same genes in an operon in both databases, 127 operons matched only partially with each other. No match was found for 140 operons in RegulonDB to those reported in ODB. Figure 4-11 gives details of the comparative study of the two databases.

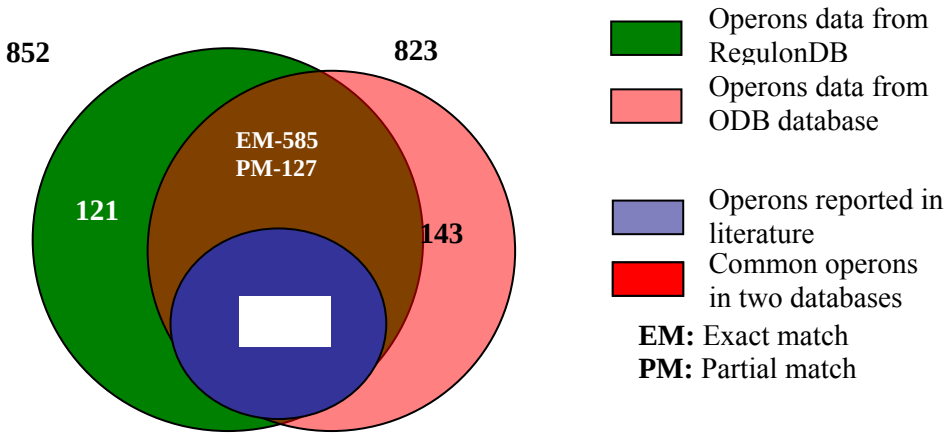


Figure 4-11 Comparison between two operon database - RegulonDB and ODB

This leads to another question, how accurate is the predicted operon data? A further analysis revealed that only 397 of 823 ODB operons are well reported in

literature (Okuda *et al.*, 2006), while other operons are predicted using various computer algorithms developed such as Ermolaeva *et al.* (2001), Salgado *et al.* (2000), Bockhorst *et al.* (2003). The predicted operons always need an experimental verification to confirm the accuracy of the prediction.

4.4.3 **Comparison between number of subsets and operons**

Genes show clustering on the genome in the form of operons. In *E.coli*, operons consist of a variable number of genes. It is possible to have one or more structural genes in an operon.

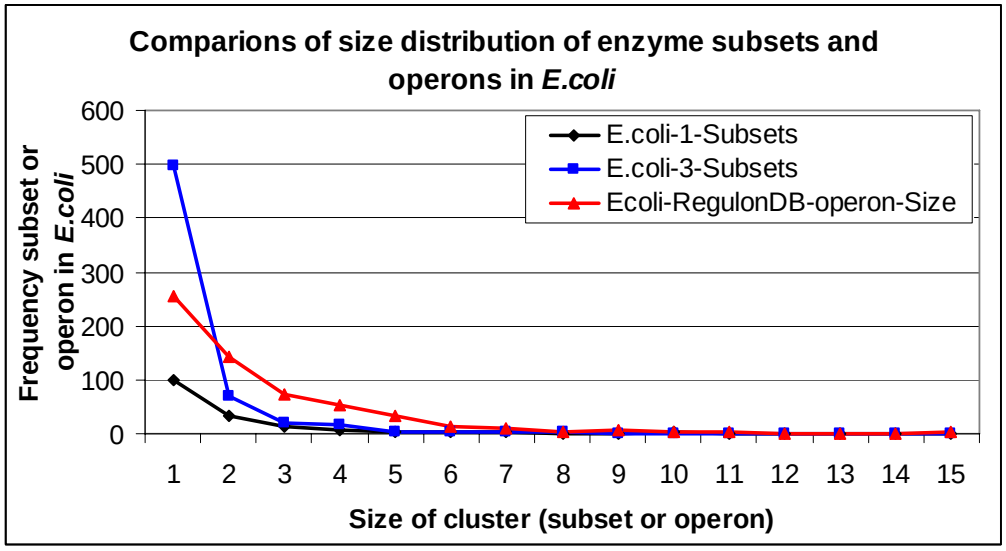


Figure 4-12 Comparison between the number of subsets and operons

To check the correlation between the size of operon (i.e. number of genes in operons) with the size of enzyme subsets (i.e. number of reactions in a subset), RegulonDB operon data (Salgado *et al.*, 2004) and the enzyme subset data from *E.coli*-1 and *Ecoli*-3 models were used (Figure 4-12). It is worth mentioning that RegulonDB shows that around 25 percent of the total operons in *E.coli* contain only one (structural) gene, suggesting that even enzyme subsets with a single reaction are equally important in understanding the correlation with operons and enzyme subsets[39].

[39] Earlier, enzyme subsets with one reaction were considered as trivial and only subsets with more than 2 reactions were considered to be significant.

Table 4-7 Enzyme subset and operon correlations in *E.coli*-3 model

Model	Total subsets	Operon Database	Matching criteria (e.g. operon to subset)	Matching criteria (Exact/partial)	Number of matches
Ecoli-3	620	RegulonDB	Subset to operon	Exact	118
				Partial	331
		ODB	Subset to operon	Exact	100
				Partial	332
		RegulonDB	Operon to subset	Exact	80
				Partial	270
		ODB	Operon to subset	Exact	67
				Partial	253

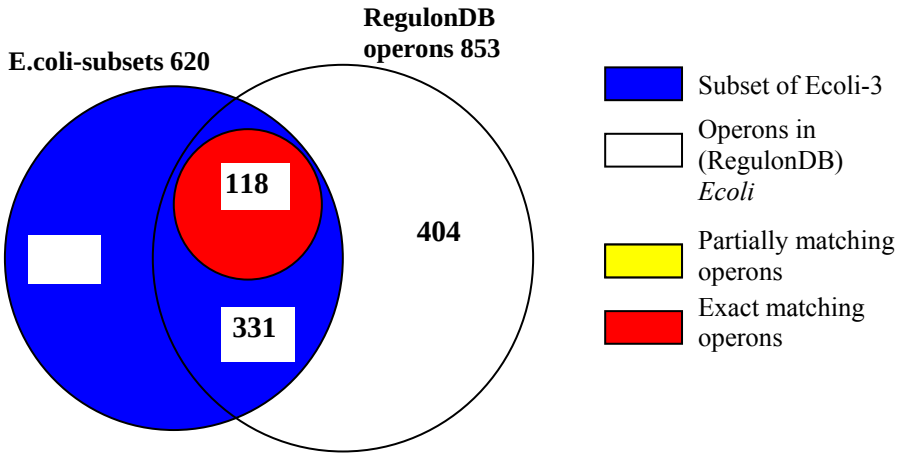


Figure 4-13 Venn diagram for subset to operon match (RegulonDB)

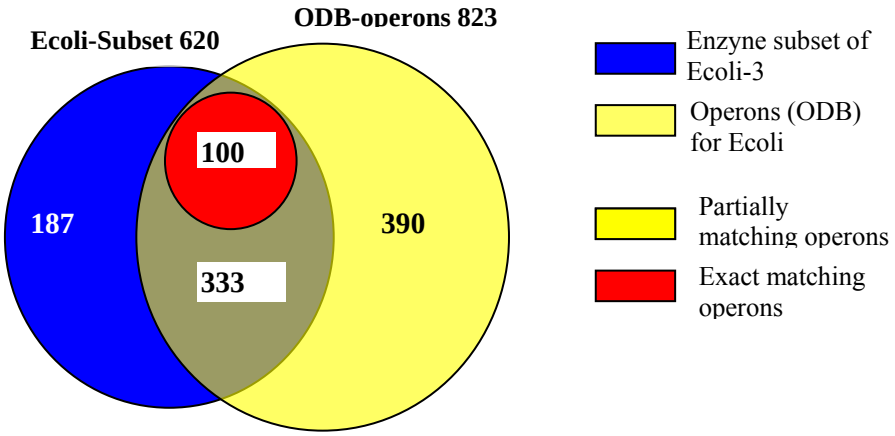


Figure 4-14 Venn diagram for subset to operon match (ODB)

E.coli-3 model enzyme subsets were converted to genes association based on the gene-protein-reaction (GPR) associations as described in section 4.1.3. An automated matching of genes in subset and operon was performed. Since this matching was of 'many to many' relationships, the results were collected into

two sets: matching of enzyme subsets to operons as a first set, and matching of operons to enzyme subsets as the second set. Table 4-7 shows the results of the matching performed on the enzyme subset and operon data and Figure 4-13 and Figure 4-14 shows the Venn diagram of the results for subset and operon match with RegulonDB and ODB, respectively.

4.5 Discussion

4.5.1 Model validation

When modelling a large genomic scale network, there is always the question of completeness and validity of the model. As discussed in Chapter 3, since there are still challenges in the model reconstruction and data representation, a model will, in a true sense, always be incomplete.

To date, no genome has been fully and accurately annotated. With a continuous sequencing and re-annotation of genomes, the knowledge of the genes is increasingly more accurate. It may be interesting to note that there are genes, whose function is still not known. In the case of *E.coli*, the genome was first sequenced in 1997 (Blattner *et al*., 1997) and was again re-annotated recently (Riley *et al*., 2006) to show that only 57.1 percent genes are experimentally verified annotations while 32 percent are computationally predicted and the rest of the 13.9 percent genes are without any functional assignment. No organism has had its biochemical reactions completely characterised. In such cases, structural modelling techniques may provide a helping hand in validation of large genomic scale metabolic models, though the constant update of the model is required as more information is revealed on the genome, proteome and reactome of the model organism.

4.5.2 Dead reactions, dead-end and orphan metabolite study

For the present study, identification of dead-end and orphan metabolites and analysis of dead reactions was performed as a measure of model validation. The above mentioned techniques assured valid structural model definition for further analysis of structural aspects such as enzyme subsets.

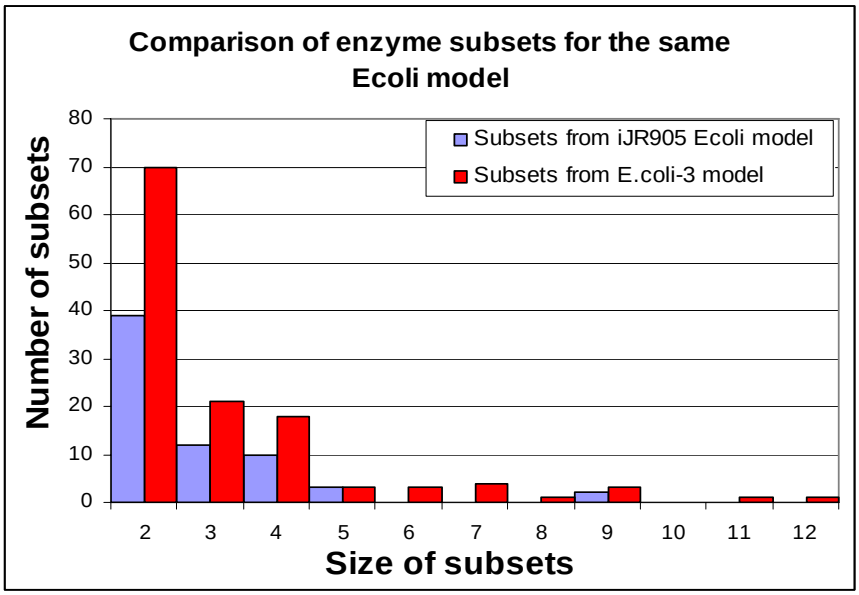


Figure 4-15 Subset comparison between iJR*E.coli*905 and *E.coli*-3 model

While the same metabolic model definitions (*E.coli*-3) were used, a significant difference was observed in the enzyme subset analysis and dead reaction analysis between the two different modelling approaches. Reed *et al*. (2003) used the same model for enzyme subset or reaction set analysis. Their analysis resulted in only 66 sets of reactions with a size of more than one reaction while using the same model, the present study showed more than 120 sets of size of more than one reaction. This is shown in Figure 4-15. This result could be explained easily on the basis of the number of dead-end and orphan metabolites and dead reactions in the model. When the *E.coli*-3 model was taken from Reed *et al*. (2003), the model showed more than 70 reactions as dead; there were many dead-ends or orphans in the model resulting in the reactions being dead, and this reduced the total number of sets in the system.

The presence of dead-ends, orphans and dead reactions in a network helps in model validation.

4.5.3 **Enzyme subset analysis**

Analysis of subsets in a genomic scale model allows the identification of substructures of the large and complex metabolic network for a given organism. While computation of elementary mode analysis is still practically impossible

for large 'omics' scale metabolic networks, the enzyme subsets do provide insights on the building blocks of elementary modes, as any elementary mode in a model is a combination of the enzyme subset of the metabolic network. A comparison between the size of reaction subsets and operons showed a similar size distribution of operons in *E.coli.*

4.5.4 **Subset as operon concept**

For the *E.coli*, enzyme subsets show similarity between the operon on the genome level. More than 15 percent of the subsets show exact matches with operons on the genome. 50 percent of subsets show only partial matches with operons in *E.coli*. This may be due to the fact that regulatory genes in the operons are not considered in the metabolic network.

Another possible reason may be the breakdown of large reaction sets to small size subsets due to the branch points in structural modelling. One of the main problems in large structural metabolic networks is the breakdown of the large subsets due to more branching. This is a well known problem in analysing large metabolic networks. As the level of network annotation increases, the complexity of the network also increases. Due to the increase in connectivity of nodes, large clusters break down to small clusters.

4.6 Challenges and problems in the reconstruction of metabolic networks: result and discussion

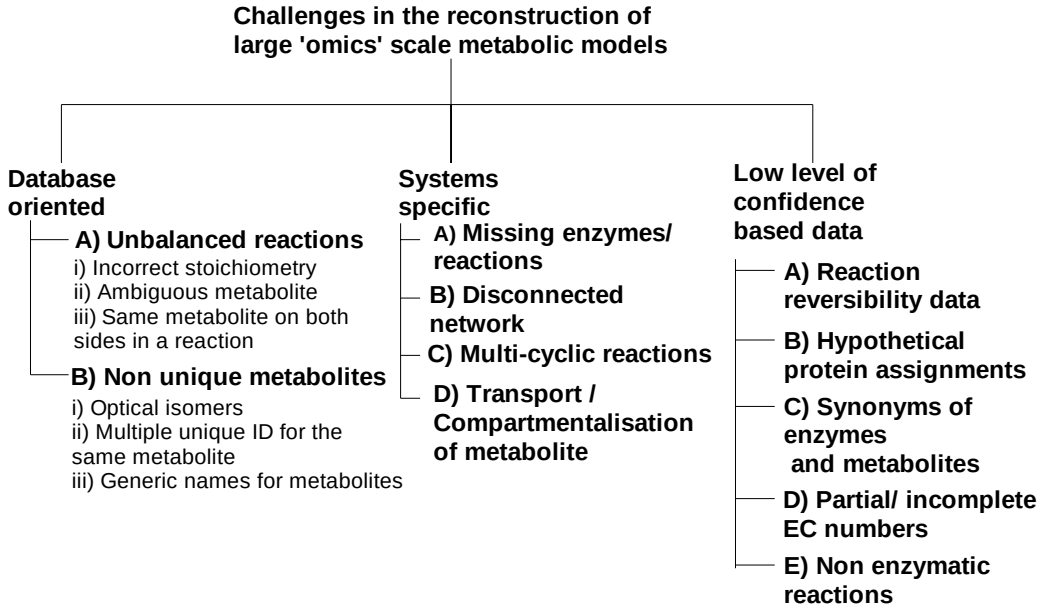


Figure 4-16 Overview of the problems in reconstruction of large 'omics' scale structural metabolic models

The reconstruction of 'omics' scale metabolic models is a data driven process and, as usual, the quality of the model is directly dependent on the quality of the data in the database. During the present study, an automated reconstruction of the *E.coli* metabolic network was tested for speeding the process of reconstruction. Organism specific databases, such as EcoCyc provide the exhaustive details on metabolic, genomic and regulatory interactions specific to the organism.

The present work of analysing the challenges in automated reconstruction of metabolic model was performed (collaboratively within the CSM lab) using various model organism specific databases available in MetaCyc and KEGG databases (Poolman *et al.*, 2006).

The problems involved in the reconstruction can be divided into three different classes:

- **Database specific problems;**
- **Systems level problem**;

- **Problems due to low level confidence based data in the database**.

4.6.1 Database specific problems

Database specific problems arising in the reconstruction are due to wrong or missing information in a database itself. Such problems can be further classified into:

A. Unbalanced reactions
B. Non unique metabolites in the reaction
C. Transport reaction with same metabolites on both sides

A. Unbalanced reactions

In the metabolic reaction database, many reactions are not stoichiometrically balanced reactions. Such errors can be further categorized as:

- **Incorrect stoichiometric reaction**

This type of error is due to the missing chemical components in a reaction or due to the missing stoichiometric coefficients of the substrate in a reaction. Both errors can cause further 'systems' level errors and may result in the reaction being dead at steady state. Eliminating such errors in the database itself is the best solution. For identifying the identification of such errors, the number of carbon, nitrogen and oxygen elements check can be performed on both sides of the substrates in each reaction. An example already discussed earlier in section 4.2.2, in which reactions were found to be dead due to incorrect and missing metabolites in the reaction.

- **Ambiguous metabolite in reaction**

When a specific substrate in a reaction is unknown or if it is not yet identified with experimental verification, databases report them with some ambiguous names in a database. Since the substrate name does not qualify as a valid chemical entity, it leads to the problem of ambiguous metabolites in the model.

For e.g. in the E.coli-4 model few reactions consists of ambiguous names as follows

RXN0-2584: "DNA with uracil due to misincorporation or deamination of cytosine" -> "DNA with uracil cleaved out leaving an AP site"

RXN0-2601: "damaged DNA pyrimidine" -> "DNA with AP (apyrimidinic site) as part of base excision repair process"

The metabolites in such reactions do not qualify as a substrate and results in further errors in model definition.

- **Same metabolite on both sides in a reaction**

Some reactions involve polymeric molecules as substrates (e.g. glycogen and DNA). The exact reaction stoichiometry of such reactions cannot be represented in a database. To overcome this issue, few reactions are reported with the same polymeric molecules on both sides since the loss of a monomer does not affect the polymer description. For instance, glycogen synthesis reaction (EC-2.4.21) involved the following form of reaction

1,4-α-D-glucan + ADP-D-glucose = ADP + 1,4-α-D-glucan

The above reaction can be represented correctly as,

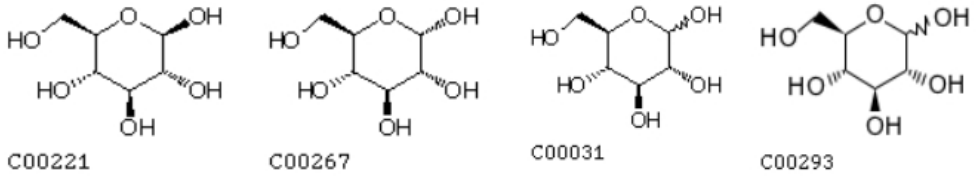
$(1,4\text{-}\alpha\text{-D-glucosyl})_n$ + ADP-D-glucose = ADP + $(1,4\text{-}\alpha\text{-D-glucosyl})_{(n+1)}$

Though this representation is stoichiometrically more accurate, it also creates problems during the model reconstruction. This creates a need to define one of the polymeric metabolites as an external metabolite.

B. Non unique metabolites

- **Optical isomers**

Various optical isomers of a compound are identified as separate entities in many databases. Assigning separate IDs for different optical isomers is worthy and correct, since it helps in defining the specificity of the enzyme. The problem still persists, when the same enzyme act on both the isomeric substrates. For e.g. three different ID's in the KEGG database represent D-Glucose, α-D-glucose, and β-D-glucose. If an enzyme is non-specific, it can act on any one of the three forms. This creates a tough task of selecting one of the three as substrate for the reconstruction of the metabolic model from the database.



IUPAC Name:
C00221: (2S,3R,4R,5S,6S)-6-(hydroxymethyl)oxane-2,3,4,5-tetrol
C00276: (2S,3R,4R,5S,6S)-6-(hydroxymethyl)oxane-2,3,4,5-tetrol
C00031: (3R,4R,5S,6S)-6-(hydroxymethyl)oxane-2,3,4,5-tetrol
C00293: (3R,4R,5S,6S)-6-(hydroxymethyl)oxane-2,3,4,5-tetrol

Figure 4-17 Glucose isomers with different Kegg IDs in the Kegg database.

- **Multiple IDs for the same metabolite**

Such errors occur in the database due to the assignment of two different unique IDs to the same chemical entity. Since chemical substances can be written by two different names e.g. glucose1,6-diphospate or glucose1,6-biphosphate, the two chemical names get two separate unique IDs. This leads to an error in the database.

In the KEGG database, as shown in Figure 4-17 different IDs were found for α-D-glucose (C00267), β-D-glucose (C00221), D-glucose (C00031) and glucose (synonyms-dextrose) (C00293) even though the last two entries may represent the same chemical compound inside the cell and can be utilised in a reaction which is non specific for glucose isomers. If different notations were used in the model to consider the enzyme specificity for the chemical isomers, then additional reaction(s) should also be added to maintain the connectivity between them. In such cases, other non-specific enzymes acting on few or all isomers in the model definition must also be addressed.

- **Generic IDs for metabolites**

In reaction databases, some reactions consists of a generic term for non specificity of the substrate, for e.g., in the EcoCyc, alcohol dehydogenase (EC-1.1.1.1) is assigned to catalyse following reaction

$$\text{An alcohol} + \text{NAD} \leftrightarrow \text{An aldehyde or ketone} + \text{NADH}$$

Another frequently observed problem in this category includes reactions such as dihydrodipicolinate reductase (EC-1.3.1.26)

2,3,4,5-tetrahydrodipicolinate + NAD(P) ↔ 2,3dihydrodipcolinate + NAD(P)H

Such reactions pose a greater threat to the metabolic modelling process as the non-specificity of the metabolites leads to errors in stoichiometry matrix with an addition of two new metabolites as NAD(P) and NAD(P)H. Whenever possible, such generic terms must be replaced with a specific chemical entity depending on the physiological information of the metabolism.

C. Transport reaction with same metabolites on both sides

This subclass should not be confused with that discussed as 'same metabolites on both sides of the reaction' in the earlier section of unbalanced reaction. Transport reactions are involved in transportation of same metabolites across various internal compartments inside the cell. Since the same (internal) metabolite is transported across the membrane, the correct description of the reaction in the database leads to a 'systems' level error by representing the same metabolite on both sides in the model definition. This further results into wrong stoichiometry of the system as described in section 2.5.3. A few databases do provide additional information on the compartment of the substrates to aid the problems due to such reactions.

4.6.2 Systems specific problems

A. Missing enzymes/ reactions

Since the reconstruction of the metabolic network is a data driven process, it is possible to see some gaps or missing data at the systems (or model) level due to lack of source information. Such gaps are possible with the models based on genome sequences. Despite the exponential increase in the gene sequencing, many genes are still are not assigned a function. Also as discussed earlier, few transport reactions are physiological processes that are mediated non-enzymatically. The lack of information on the spontaneous or non-enzymatic reactions that may not be available during the model reconstruction results into gaps and dead-ends in the metabolic network.

B. Disconnected network

In case of some parasites, it is known that their metabolic network may be a disconnected network as they can utilise host metabolites (and some of the host biochemical reactions) for their survival. However, in case of free-living organisms such as many plants, animals and prokaryotes, it is unlikely that the network is disconnected, although there is no such documented proof available in literature. This assumption is based on the fact that for a free-living organism, nutrition and growth are only due to utilisation of the carbon, nitrogen and oxygen; under these conditions, one should find the network continuous since all the internal substances are produced chemically from the nutrients taken up by any cell. For the present study, the disconnected networks were considered as 'systems level' problems suggesting missing links in model definition.

C. Orphan and Dead end (unbalanced) metabolites

As discussed earlier in Section 2.3.3, orphans and dead-end metabolites are one of the main causes of reaction(s) being dead in the system. Identification of orphans is done by finding the connectivity of each metabolite in the model. The orphans have a connectivity score of '1' in the model. The dead-end metabolites are much more difficult to be identified by just the connectivity score. To a certain extent, identification of some of the dead-end metabolites is possible with the null space and enzyme subset analysis. Two such cases are discussed in Chapter 4 with respect to the *E.coli* genomic-scale model.

D. Multi-cyclic reactions

Reactions involved in the synthesis of polymeric molecules pose a problem of correct reaction stoichiometry in genomic scale models. For the modelling purpose, manual corrections are needed for such reactions either by defining the polymeric molecule as external or by giving an approximate stoichiometry of the monomers in the reaction. Similar problems are observed in fatty acid synthesis, wherein the same set of reactions (or enzymes) acts on different substrates for the synthesis of fatty acids.

4.6.3 Problems due to low level confidence based data in the database

The data derived from predictions rather than experimental verification poses a new challenge when such data appear in the database. Since such data may need further experimental verification, its inclusion may create serious problems in the metabolic modelling process. The following cases account for the common errors in the metabolic modelling.

A. Reaction reversibility data

Most of the reactions are mediated by enzymes in the cell that are considered as either reversible or irreversible. It is extremely important to know the preferred direction of the reaction in a particular organism. While measurement of the preferred rate of reaction flow is impossible in all the organisms, the reversibility criteria for an enzyme is generally assigned from the literature with known enzyme kinetics in any one species. Such assignments may not always be correct across the organisms.

B. Putative protein assignment and orphan enzymes

Although the similarity or homology based search is a very potential tool in gene assignment across the organisms, it fails to determine the functions for many genes and produces incorrect or error prone annotations creating the hypothetical protein problem (Bork & Koonin, 1998). Using information based on such assignment may lead to incorrect model definition.

Those proteins that have been well studied experimentally but are not yet correlated to the genome sequence data in current databases are called orphan proteins or enzymes. Naumoff *et al*. (2004) observed that such orphan enzymes are sometimes related to wrong annotation. The gene encoding putrescine carbamoyl transferase (EC-2.1.3.6), for example, was wrongly annotated across several completely sequenced organisms as ornithine carbamoyltransferase (EC 2.1.3.3) (Lespinet & Labedan, 2005).

C. Synonyms of enzymes and metabolites

Most of the chemical entities, substrates and enzymes have many synonyms. This leads to problems in collecting and validating of metabolic models from various databases. For instance, glucose can be represented by various different names across the databases such as glucose, D-glucose, α-glucose or by its IUPAC name. While dealing with large 'omics' scale data mining for model validation, such synonyms pose a tough challenge for the user.

The worst case is for enzymes with no EC numbers. In such case, if the enzyme acts on different substrates, it is possible that it has more than one name, as the names of the enzymes are mostly given by its action and specificity for the substrate. In addition, multifunctional but identical enzymes are difficult to identify, if they are known by various synonyms across databases or literature.

D. Partial/incomplete EC numbers

Green and Karp (2005) observed that in the KEGG and other databases EC 4.2.1.- enzyme was linked to 16 reactions. Two genes b0036 (Carnitinyl-CoA dehydratase EC 4.2.1.-) and b1517 (Putative aldolase yneB EC 4.2.1.-) were both assigned to catalyse a set of 16 distinct reactions. According to EcoCyc, b0036 and b1517 encode a carnitine racemes (EC 5.-.-.-) and a putative aldolase, respectively. The function of b0036 is supported by experimental evidence, while b1517 is assigned as putative. The wrong assignment in KEGG was propagated across many other metabolic databases.

Another minor but significant problem regarding enzyme classification is that the class or subclass of proteins which are still not classified or confirmed by the EC committee are kept under a temporary class or subclass or sub subclass with numbers 98 or 99 (e.g. EC 1.2.98.- or EC 1.99.2.-). Such enzymes get a new entry over a period of time, causing the problem of new EC number assignment. Since not all the databases update the EC number, such changes may result in wrong assignment of genes and reactions in the database (Green & Karp, 2005). During the model reconstruction, the change in the assigned unique EC number

created problems for further gene-protein-reaction association relationship as explained in Section 3.3.4.

4.6.4 Gene-protein-reaction association oriented problems

A. Inconsistency in the gene names

Like metabolites, genes too suffer from the problem of synonyms. The same gene can be referred to by different names across the databases and literature, creating confusion during the automated GPR associations. Constant update in the synonym of genes leads to errors during the model interrogation and analysis. To avoid such confusions, additional information on the position or location of the gene on the chromosome is necessary to identify the same gene(s). More precisely, the information on start codon and stop codon is needed.

E.g. the gene 'tdcG' coding for L-serine deaminase-3 has three Blattner IDs as b4471 , b3111 , b3112 and four gene names as 'YhaQ' , 'YhaP' , 'SdhY', 'TdcG'. The problem becomes much worse when the expression of these genes are analysed from microarray data for expression profile or coexpression. The user needs to be very cautious about such genes, and information about other synonyms is very important in such cases.

B. Complex gene protein associations

As discussed in section 4.1.3 of this Chapter, the many to many relationship between genes, proteins and reactions results in to a highly complex gene protein reaction associations.

Another problem includes enzymes for which the gene is not yet identified, e.g. the enzymes EC-3.2.2.14, EC-3.2.2.10, EC-2.7.8.1, EC-5.1.99.1, EC-3.3.1.1, EC-2.7.1.100, EC-5.3.1.23, EC-3.1.3.4, EC-3.1.3.7, EC-3.5.1.42, EC-3.2.1.26, EC-2.6.1.2 are well characterized for carrying reactions in *E.coli* but genes encoding the above enzymes are yet to be found on the *E.coli* genome. For such enzymes, links between the gene-protein-reaction associations are still missing and further research is needed to identify encoding genes for such enzymes by experimental and computational tools.

C. Non enzymatic reactions

As mentioned earlier, there are a few reactions that take place spontaneously. In addition, non-enzymatic reactions include passive or facilitated transport of substrates *via* cell wall membrane proteins. Since such reactions are independent of the transcriptome, they need manual intervention or physiological or biochemical knowledge to be included into the model during reconstruction.

Non-enzymatic reactions can not be directly associated to the genome of the organism. These reactions are not directly controlled by the genome. Nevertheless, they may be indirectly controlled by other neighbouring reactions (genes) responsible for the production of the substrate metabolites of the non enzymatic reaction. Understanding of such reactions in the system plays a vital role in filling the gaps in metabolic model definitions. Certain transport reactions allow free diffusion of the substrates across various compartments making the substrate always available for various other reactions.

One such example is the reaction carried out by the enzyme 6-phosphogluconolactonase,

6-phospho-D-glucono-1,5-lactone + H2O → 6-Phospho-D-gluconate + H

This reaction, 6-phosphogluconolactonase (EC-3.1.1.31) of pentose phosphate cycle can also proceed spontaneously, so the physiological role of the enzyme is not understood, though the gene responsible for the enzyme is known in *E.coli* mutant strain as ybhE (b0767) (Thomason *et al.*, 2004).

Another example is a reaction from leucine biosynthesis, where 2-isopropyl-3-oxosuccinate spontaneously converted to 2-ketoisocaproate and carbon dioxide.

2-isopropyl-3-oxosuccinate → 2-ketoisocaproate + CO_2

D. Genes with unassigned function

To date, more than 200 genomes were completely sequenced and are now available for study. However, no single genome is fully and accurately annotated, and functions of many genes are yet to be identified on such well sequenced genomes.

The problem associating the function to such orphan genes in context to metabolic pathways was tackled by various methods. One approach used comparative genomics to find the similar functional class of genes (Osterman & Overbeek, 2003). Such methods suggest possible hits of functions of some genes, but for few others identification of functional assignment can not be possible or detected experimentally.

E. Database issues and data handling

Since there are several databases for genome and metabolic reactions for various organisms, the last problem is how to handle such diverse and huge data to obtain desired information. Various databases use their own unique IDs, and different synonyms for the same entry. This poses another challenge of validation of the information obtained from one database to another.

Another problem is the replication of one database error across others. Most of the databases automatically update their content from other similar open source databases. This causes propagation of an error in one database to many other similar databases. All databases are constantly updated with the addition of new or correct information, making older versions redundant and useless. Hence, the models need to be reconstructed with constant updates so as to make it more accurate.

4.6.5 *Ecoli*-4 model enzyme subset analysis

E. coli-4 model which was build directly from the EcoCyc database consists of large number of dead reactions, dead end metabolites and orphan metabolites. This model was further studied for enzyme subsets, presence of dead reactions and orphan and dead-end metabolites.

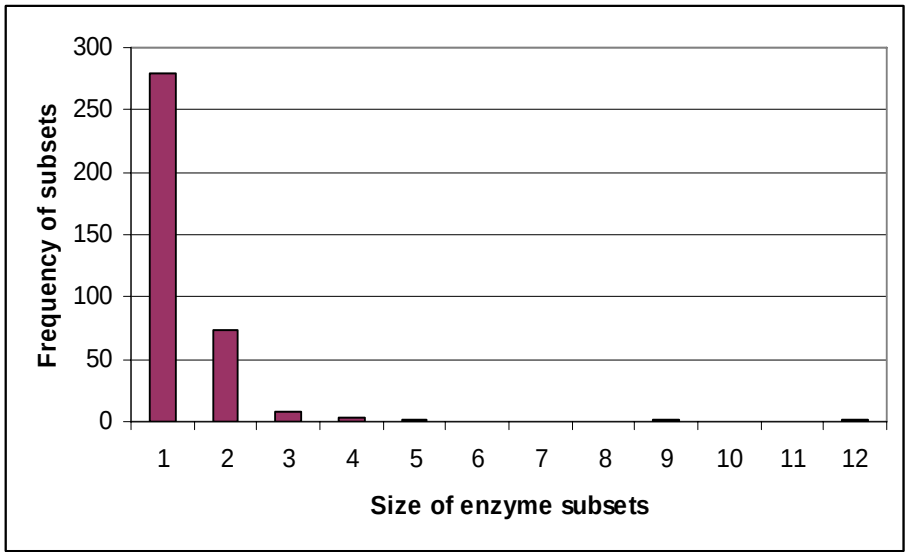


Figure 4-18 Enzyme subset vs. frequency of subsets for *E.coli*-4 model.

The enzyme subset analysis for *E. coli*-4 model shows more than 40 percent of the reactions to be dead in the model. A dead subset of 753 reactions suggests that the model to consist of dead-end and orphan metabolites. Even after defining a few orphan metabolites as external, the model still suffers from the various problems discussed in Chapter 3.

Figure 4-18 shows the size distribution of the subsets in the *E.coli*-4 model. Due to the problems of missing reactions (links) and same metabolites on both sides of the reaction, this model was not considered in the further study.

4.7 Conclusion

In conclusion, reconstruction of a genomic scale metabolic network from publicly available databases poses totally different challenges. Though in most of the cases, the data representation of metabolism at database level is correct, at systems level it poses a new challenge.

An automated reconstruction of structural metabolic model can be possible from the large metabolic, genomic and organism specific databases. However, such reconstruction suffers from various problems discussed in section 4.6 and despite the large amount of data, needs a laborious manual curation of the model.

Enzyme subsets represent the substructure of the metabolic network. Reactions in a subset show all or none relationship to fluxes of the network. This unique property of the subsets appears to match with operons in *E.coli* on genome scale. Enzyme subset analysis can be applied to large omics scale models. However, elementary mode analysis can not be successfully applied due to the combinatorial explosion. Enzyme subsets provide information on the building blocks of the elementary modes and help in understanding the substructure of the metabolic network.

In reality, irreversible reactions in a subset must follow same direction to qualify as set. However, in theory, null space ignores the reaction directionality. Idenification of some complex dead-end metabolites can be possible with enzyme subset analysis with a further reaction directional contstraint.

Chapter 5
Gene expression relationship and metabolic network substructure

5.1 Introduction

The rapid expression detection technique called gene expression microarray analysis provides high throughput measurement of relative mRNA levels for thousands of genes in a biological sample (Schena *et al.*, 1995). Many laboratories have collected and analysed microarray data (Werner-Washburne *et al.*, 2002), and made it publicly available (e.g. Stanford microarray database (Ball *et al.*, 2005)) or on researchers' web sites[40]. These resources which were initially kept for the cross verification of the published results, have also served as raw data for further analysis, that went beyond the scope of the original experiments (Lee *et al.*, 2004).

5.1.1 Microarray technology

A. Basics of microarray

The use of microarray for gene expression profiling was first published in 1995 (Schena *et al.*, 1995). DNA microarrays (also known as gene chip, DNA chip, or biochip) use an array of cDNA strands printed on a solid support to detect mRNA expressions in a cell. This technology was evolved from Southern blotting, where fragmented DNA is attached to a substrate and then probed with a known gene or fragment. Such analysis is called expression analysis or 'expression profiling' for a given biological sample. Due to the technological advances, microarray chips can be labelled with tens of thousands of distinct fragments. This allows the detection of a large number of gene expressions in a short time. DNA microarrays therefore dramatically accelerated many research techniques including disease gene identification (Warrington *et al.*, 2002), drug research and genetic manipulation.

[40] For example, http://bioinformatics.upmc.edu/Help/UPITTGED.html

Depending on the experimental design and platform, the microarray chips are of two main types, cDNA arrays, where polymerase chain reaction (PCR) products are used as features or spots and oligonucleotide arrays (e.g. commercially available Affymetrix gene chip) where short (20-80 base pairs) oligonucleotide sequences are used as spots.

C. Applications of microarrays

- Microarrays provide the expression data for thousands of genes for a given biological sample.
- They help to determine gene expression profiling or measurement of the amount of mRNA expressed in a sample.
- They also play important roles in gene discovery, diagnosis of diseases like cancer, and help in identifying the response of expression of genes by certain drugs in the body, thus playing a significant role in drug discovery process (Meloni *et al.*, 2004).

D. Challenges in microarray data analysis:

Despite the high throughput study of gene expression, microarray techniques pose a new challenge of analysing the huge quantity of expression data obtained from various experiments (Nadon & Shoemaker, 2002). Application of various statistical methods such as linear regression, data normalisation, analysis of variance, covariance and correlation coefficient analysis is needed to avoid false negative and false positive associations from the final data (Causton *et al.*, 2003; Li, 2006). To a certain extent, applying statistical tools identifies such artefacts and helps in arriving at proper conclusions (Draghici *et al.*, 2006).

5.1.2 Basics of statistical testing:

A. Correlation coefficients

Correlation analysis determines the extent to which changes in the value of one variable is associated with changes in another variable. The data for a correlation analysis consists of an expression vector for two genes (e.g.

expression of a gene A in 100 different microarray experiments in one column and expression of gene B in the same 100 samples in second column). A correlation coefficient provides measures of how the expression of two genes is related in the multiple samples.

Pearson's correlation coefficient

The Pearson's correlation coefficient (denoted by R or ρ) measures the strength of the linear relationship between two variables. The value of R can range from -1 to 1. At the extreme (i.e. by ignoring the negative sign), Pearson's correlation coefficient of 1 indicates strong association between the two variables (i.e. perfectly correlated variables). It also suggests that the change in one variable can also be predicted accurately from the other variable. Values near zero imply an absence of any correlation between the two variables (Sokal & Rohlf, 1995). Pearson's correlation coefficient for two genes (u, v) in m experiments is given by;

$$\rho(u,v) = \frac{covariance\ (u,v)}{\sigma_u \sigma_v} \tag{11}$$

where σ is the standard deviation for a given gene from m number of samples. Covariance and standard deviation can be calculated as,

$$covariance\ (u,v) = \frac{\sum_{i=1}^{m}(u_i - \bar{u})(v_i - \bar{v})}{(m-1)} \quad and\ \sigma_u = \frac{\sqrt{\sum_{i=1}^{m}(u_i - \bar{u})^2}}{(m-1)} \tag{12}$$

where $\bar{u}$, $\bar{v}$ is the arithmetic mean of the expression values from m experiments for gene u and v, respectively.

In the context of microarray data analysis, two genes with different levels of expression but 'parallel' expression patterns would be considered as closely related. As shown in the Figure 5-1 scatter plot, if the two expression vectors show a scatter characteristic of correlation coefficient close to (+)1, then it suggests a strong positive correlation (shown with blue dots and pink shaded area on the graph in Figure 5-1). This hints that an increase in the expression of one gene can be related to an increase in the expression of the other gene. Genes with a negative (close to -1) correlation coefficient show 'perfect negative' or

'anti correlations', suggesting that an increase in the expression of one gene can be related to the decrease in expression of the other (shown with pink dots and green shaded area in the graph in Figure 5-1) (Causton *et al.*, 2003). Genes with Pearson's correlation value near 'zero' suggests weak correlation or unpredictable correlation between the two vectors (shown with red triangle dots in the centre of the plot in Figure 5-1).

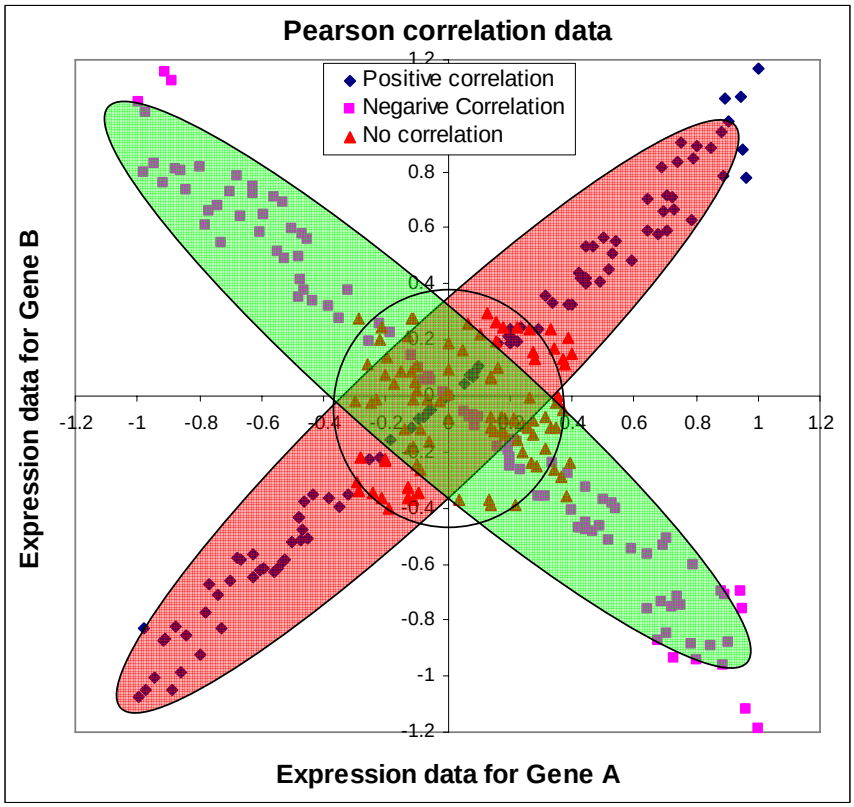


Figure 5-1 Significance of Pearson's correlation coefficient for gene expression

The two other commonly used correlation coefficients include Spearman's correlation coefficient and Kendall's (τ) correlation coefficient (Causton *et al.*, 2003).

B. Hypothesis testing

While comparing results of multiple samples, most statisticians mainly worry about two kinds of error rates, type I error (α) and family-wise error in such multiple comparisons.

- **Significance level:**

The significance level (donated by α for type I error or β for type II error) for a statistical hypothesis test is a fixed probability of wrongly rejecting the null hypothesis, despite the fact that it is true (Bland, 2001).

- **Family-wise error rate**

While performing a series of significance tests, the family wise error rate (FER or FWE) is the probability that one or more of the significance tests results gives a type I error (Sokal & Rohlf, 1995).

- **P-value**

A P-value is a measure of the probability or chance of not getting the desired result by chance, if the null hypothesis were true (Dallal, 2004). If the fixed significance level (e.g. 0.05 for 5% confidence) is greater than or equal to the p-value, one can reject the null hypothesis. If the fixed level value is less than the actual p-value, it suggests failure to reject the null hypothesis or accepting the null hypothesis.

C. Test for multiple comparisons:

1. Bonferroni-Holm test

The Bonferroni-Holm step down test can be applied to any family of pair wise comparisons as it does not require any assumption. (Hommel, 1988). It is a highly conservative test; its family-wise error does not exceed 'alpha'. The following procedure was adopted for Bonferroni-Holm test (Sokal & Rohlf, 1995).

If the data consists of 'n' number of pairs with 'R' values with corresponding 'p-values', a modified 'alpha' (i.e. significance value) was obtained by dividing 'alpha' by (n-j) where j is the rank of the pair after arranging the pairs in descending p-values.

E.g. if n pairs (samples) are present in the study, then R_1, R_2, …., R_n represents the Pearson coefficients and p_1, p_2, …., p_n represents the corresponding p-value of the pairs, then all the pairs were rearranged with descending order of the p-values.

If $p_1 >$ (*alpha*/n);

Then all the hypotheses were retained with the conclusion that there is no evidence of differences between the means at significance level *alpha*.

If $p_1 <=$ (*alpha*/n);

Then reject the hypothesis related to comparison R_1 with the conclusion that the means in comparison C_1 are significantly different at level alpha; go to the next to compare C_2 with [alpha/(n-1)].

……..

Continue until a pair with acceptable null hypothesis is found, then stop further testing and accept the null hypothesis for all the remaining pairs in the test.

Another common alternative test is the Sidak-Holms (SH) method, which is very similar to the BH test (Sokal & Rohlf, 1995). In this test, the ordered p-values are compared with the Sidak adjustment, $1-(1-\text{unadjusted p-value})^{n-j}$, for j=0,1, .., n-1; where n is the sample size and j is the rank of the ordered sample data. This test is slightly less conservative than BH test (Hommel, 1988).

D. Data clustering techniques

Cluster analysis is a tool to analyse data by sorting and rearranging different objects into groups of similar patterns (Eisen *et al.*, 1998). Two objects belong to the same group if they share a high degree of similarity. Cluster analysis includes a number of different algorithms and methods such as hierarchical clustering, K-means clustering, principal component analysis (PCA) etc.

Hierarchical clustering

This is an agglomerative clustering method which joins similar genes into groups. The iterative process continues until all the groups are connected to

produce a hierarchical tree (Eisen *et al.*, 1998). The decision of linking the two groups is taken by linkage method. The linkage methods can be based on the distance measure such as Euclidian distance (*TIGR MeV MultiExperiment viewer version 3.1*, 2005). Linkage methods can be further classified as,

- Single - cluster objects based on the minimum distance between them (also called the nearest neighbour rule)
- Complete - cluster objects based on the maximum distance between them (also called the furthest neighbour rule)
- Average - cluster objects based on the average distance between all pairs of objects

5.1.3 **Gene coexpression**

Gene coexpression allows identification of genes with similar expression pattern across a number of microarray experiments. One of the many uses of microarray data is to exploit the gene coexpression. Coexpressed genes are hypothesised to have a similar functional relationship(Stuart *et al.*, 2003).

Gene coexpression has been shown in several studies to always correlate with the functional similarity or relatedness (Eisen *et al.*, 1998). Stuart *et al.* (2003) also showed gene coexpression studies across various species to help in the identification of conserved genes of similar functions. In a recent study, Lee et al. (2004) analysed 60 microarray datasets using gene pairs coexpression, and showed clustering of the data to show functionally related genes with positive correlation. In both the studies, genes with higher correlation (e.g. >0.6) were hypothesized to be biologically relevant and related to similar metabolic or biochemical functions. Allocco *et al.*(2004) extended the approach of Lee *et al.* (2004) to investigate and quantify the link between co-expression and co-regulation of the genes in yeast using microarray data. Their investigation showed that genes with similar function share a common regulatory mechanism.

The microarray data was used to study the coexpressed genes that are shown to correlate with the functional relationships, mostly with the physical interactions between the encoding proteins. One major problem in such coexpressions is that

not all the relationships among the genes imply a causal relationship among the transcript level (Eisen *et al.*, 1998). The possible reasons for such a problem are the high noise level of the microarray data ranging from experimental problems, errors in hybridization of probe and sample, and reading the microarray data from the chips (Draghici *et al.*, 2006).

Despite the problems involved in the identification of the coexpression data, the importance of identifying the coexpressed genes proved vital in many studies involving organism as well as organs (Lee *et al.*, 2004). Tjaden *et al.* (2002b) used high density oligonucleotide probes to *E.coli* transcriptome and coexpressed genes. Their approach identified 317 novel transcripts from the study of 4052 coding transcripts. However, despite the sophisticated approach, it failed to identify many transcripts (Tjaden *et al.*, 2002b).

5.1.4 **Literature on gene expression and metabolic systems**

Various research studies have been performed to relate the general topological organisation of metabolic network and gene coexpression in an organism. Ihmels et al. (2004) used the reaction set for *S. cerevisiae* derived from the KEGG database to combine expression data from public repositories to identify correlations between the expression of genes from the same metabolic pathway[41]. They observed that enzymes involved in the same metabolic pathway were often co-expressed, and that the co-expressed enzymes were usually arranged in a linear pathway. The authors also found that at branch points, the incoming branch is normally co-regulated with only one of the outgoing branches, which leads to linear metabolic flow through the network. While the above study was only based on the connectivity between substrates and did not consider molecules involved in energy metabolism nor reducing cofactors, which are known to be the most connected metabolites in the networks. Incorporation of these compounds in the analysis might reveal less obvious enzyme interactions mediated by these common metabolites.

[41] The exact definition of 'pathway' is a matter of debate between various researchers and there is no set definition or criteria for defining a pathway (see Section 3.5)

Kharchenko et al. (2005) also used topological genome scale metabolic model to investigate the significance of coexpression of genes and the average network distance. They correlated the coexpression of all possible two/three reaction motifs in the metabolic network. The motifs were classified and arranged according to local structural/topological property of metabolic reactions such as convergent, cyclic, divergent, serial, parallel, etc. They observed that positive gene coexpression decreases monotonically in the order of motifs described, while negative coexpression is strongest at intermediate network distances. They observed that such basic topological motifs of the metabolic network exhibit statistically significant differences in coexpression behaviour.

Patil and Nielsen (2005) used a genome-scale model of *S. cerevisiae* to show that it is possible to reveal patterns in the metabolic network that follow a common transcriptional response and identified 'reporter' metabolites around which the most significant transcriptional changes occur. They used gene expression data from wild-type strains grown on several different carbon sources. Using the graph-theoretical representation of metabolic reactions, they identified highly regulated metabolites (reporter metabolites) and sub networks, and found that the sub networks are significantly correlated to changes in gene expression.

In the present study, the correlation between the enzyme subsets and gene expression of the genes responsible for the reactions in a subset is analysed. The aim was to test the hypothesis that, transcripts encoding enzymes in a subset share a common regulatory unit on genome scale. As discussed in the earlier Chapter, reactions in a subset show flux proportionality, it is expected that the transcripts responsible for such reactions must also show some similarity in their expression profile. The subsets obtained from *E.coli*-3 model are examined to determine the extent to which the genes for the enzymes catalyses the reactions using the *E.coli* microarray experimental data.

5.2 Methodology

5.2.1 Generation of enzyme subset data and gene-protein-reaction association

The genome scale E.coli-3 metabolic model was used to obtain the enzyme subset data (see section 4.3.2). Gene-protein-reaction associations were applied to obtain subsets with associated genes for each reaction. A subset containing dead (reactions) enzymes and subsets with only one gene encoding one enzyme[42] were eliminated from the further analysis.

5.2.2 Expression profile data for *E.coli*

A. Source of processed microarray data

The coexpression data used in the present study was obtained from Comprehensive Systems Biology (CSB) database hosted at Molecular Plant Physiology Div., Max Planck Institute, Golm, Germany (Steinhauser *et al.*, 2004b).

Table 5-1 Source of the raw microarray data used for the production of co-response matrices
Reproduced from (Steinhauser *et al.*, 2004a)

Dataset	M96A	M96B
Microarray platform	Affymetrix, Oligonucleotide Hybridization	Colour-coded EST hybridization (cDNA)
Number of experiments	16	50
Number of genes	4345	4290
Method of single	Log-transformed	Log-transformed
% of missing data	0	0
Experiment		
Growth curve	16	50
Isolation of datasets from above data		
Isolated datasets	M96Aec864	M96Bec864
Number of genes	864	864

The EcoCoR@CSB.DB[43] database contains transcriptional co-responses (gene-gene correlations) from *E. coli* K-12 (MG1655) microarray data from two publicly available microarray databases, Stanford microarray database (Ball *et*

[42] Since such subsets will contsists of only one gene, a coresponse correlation of genes can not be obtained for such subsets.
[43] http://csbdb.mpimp-golm.mpg.de/csbdb/dbcor/eco.html

al., 2005) and ASAP microarray database (Glasner *et al.*, 2006). The co-response data allows inferring hypotheses on the functional interaction of genes. The log-transformed microarray data was used to obtain the Pearson's correlation coefficient (denoted by 'R') and corresponding 'p-value' of gene pairs from all the genes in the microarray data. As shown in Table 5-1, CSB database provided two datasets (M96A and M96B). The only difference between the two data sets was experiments were performed using different types of microarray techniques. Pearson's correlation coefficient was computed to obtain the coexpression matrices. In each dataset, each gene pair consists of Pearson's correlation coefficient and observed 'p-value', datasets M96A and M96B[44] were originated from of 16 oligonucleotide array and 50 spotted array experiments, respectively. Almost all the experiments used glucose as the main carbon source (Allen *et al.*, 2003). For the present study, M96A dataset was further studied extensively but similar study can be performed on M96B dataset.

B. Isolation of coexpression co-response matrix

The reactions in *E.coli*-3 model were converted to genes *via* gene protein reaction association (refer Section 4.1.3). From the master co-response matrix (M96A) all the genes associated to reactions in *E.coli*-3 model (864 genes) were isolated to obtain a new matrix M96Aec864 with the corresponding 'p-value'.

5.2.3 Visualization and data manipulation

TM4-MeV (*TIGR MeV MultiExperiment viewer version 3.1*, 2005) software was used for visualisation of the raw and clustered coexpression matrices. This software provides a graphical user interface for visualisation as well as analysis of the data table. It was developed for microarray data analysis, but technically it can be used to analyse a large quantity of any other matrix data. The software provides 12 clustering algorithms, allows exporting images, extracting the clusters in standard 'NEWICK' tree file format. TM4-MeV is available as an open source package. For performing statistical analysis, additional python scripts were written to handle very large gene coexpression matrices.

[44] These pre-computed co-expression data matrices with Pearson's (R) correlation and corresponding 'p-value' were generously given by Prof. Steinhauser (CSB.DB). To avoid the confusion, now onwards these datasets will be called as 'master datasets'. However, same data can also be obtained freely from CSB.DB web pages.

5.2.4 Enzyme subset based clustering of gene

As shown in the Figure 5-2, the enzyme subset analysis was performed on the *E.coli*-3 model. Each subset was then converted in terms of the genes responsible for the reactions in the subset using gene-protein-reaction associations (as discussed earlier in section 4.1.3).

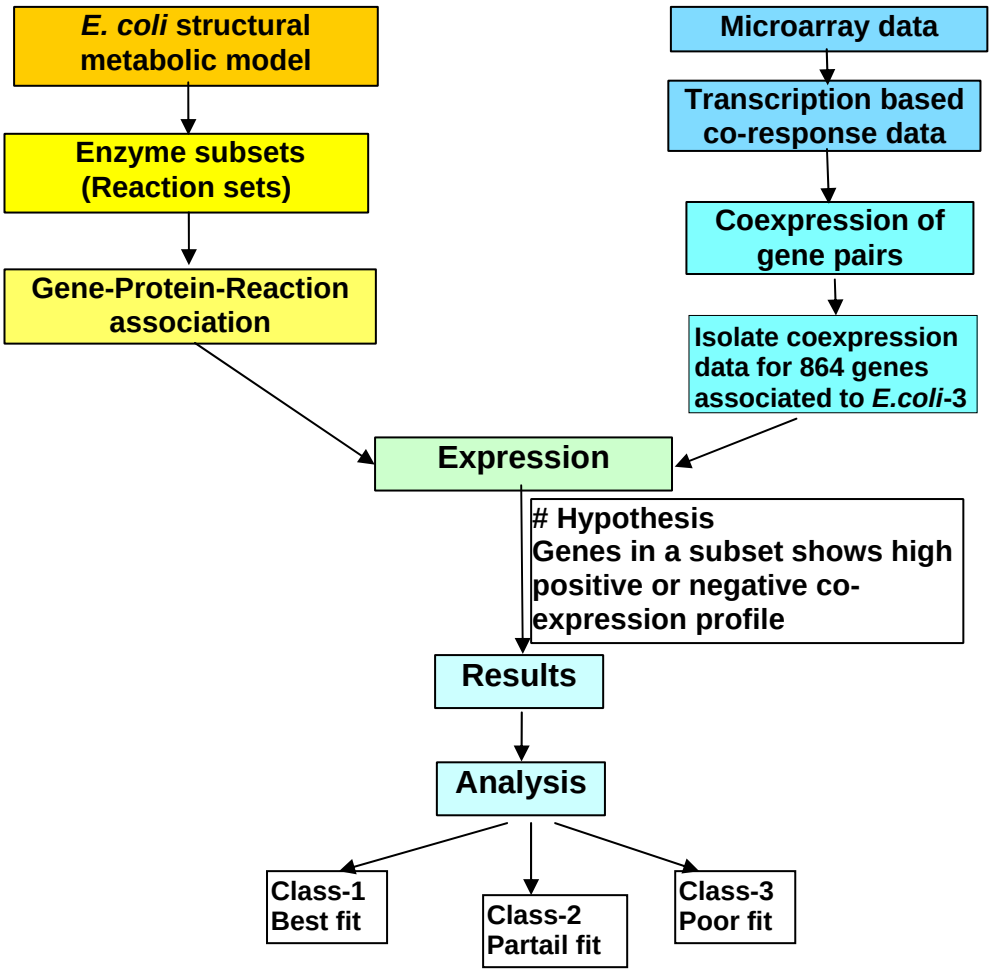


Figure 5-2 Schematic diagram of enzyme subset based gene co-response for microarray data clustering

The expression data obtained in terms of Pearson correlation coefficient for the corresponding genes in *E.coli* were then correlated to the pairs of genes in an enzyme subset (e.g. if there were three genes in a subset as geneA, geneB and geneC, then all the three pair-wise combinations (geneA-geneB, geneB-geneC and geneA-geneC) were considered). The Pearson correlation coefficients for gene pairs in a subset were extracted from the M96Aec864 and further studied

to identify the positive or negative correlations. The confidence of the correlation was tested using testing procedure (explained in the section 5.1.2) for multiple correlations.

The hypothesis was tested as follows-

Null hypothesis (H_0) : Genes in a subset do not associate functionally to each other and show coexpression by chance.

Alternative hypothesis (H_1) : Genes in a subset always show high positive or negative Pearson correlation coefficient

Sort all the gene pairs according to ascending original 'p-value' and test each Pearson's correlation coefficient of a gene pair by testing:

If, original 'p-value' > modified 'p-value' or 'alpha':
Accept null hypothesis
else:
Reject the null hypothesis

where modified 'p-value' is obtained according to the procedure explained under Bonferroni-Holm (BH) correction in Section 5.1.2.

An example of the calculation of the confidence for a pair of genes in a subset is shown in Appendix C.

5.2.5 **Traditional clustering approaches**

To identify groups or clusters of genes using traditional clustering techniques, Python based statistical clustering tool 'tree' written by Dr. Poolman (unpublished data) and built-in algorithms in MeV software such as hierarchical clustering (Eisen *et al.*, 1998) were used to produce gene clusters from the two Pearson's correlation matrices M96Aec864 and M96Bec864.

5.3 Result

5.3.1 **Traditional clustering techniques**

The raw coresponse matrix (M96Aec864) was visualised using MeV software. The genes in the M96Aec864 were arranged in the order in which they appear on the chromosome of *E.coli*. Figure 5-3 (top figure) shows the dot-plot of the M96Aec864 matrix, where the spot is green if Pearson's coefficient is -1, red if the value is 1 and black indicating value of 0.

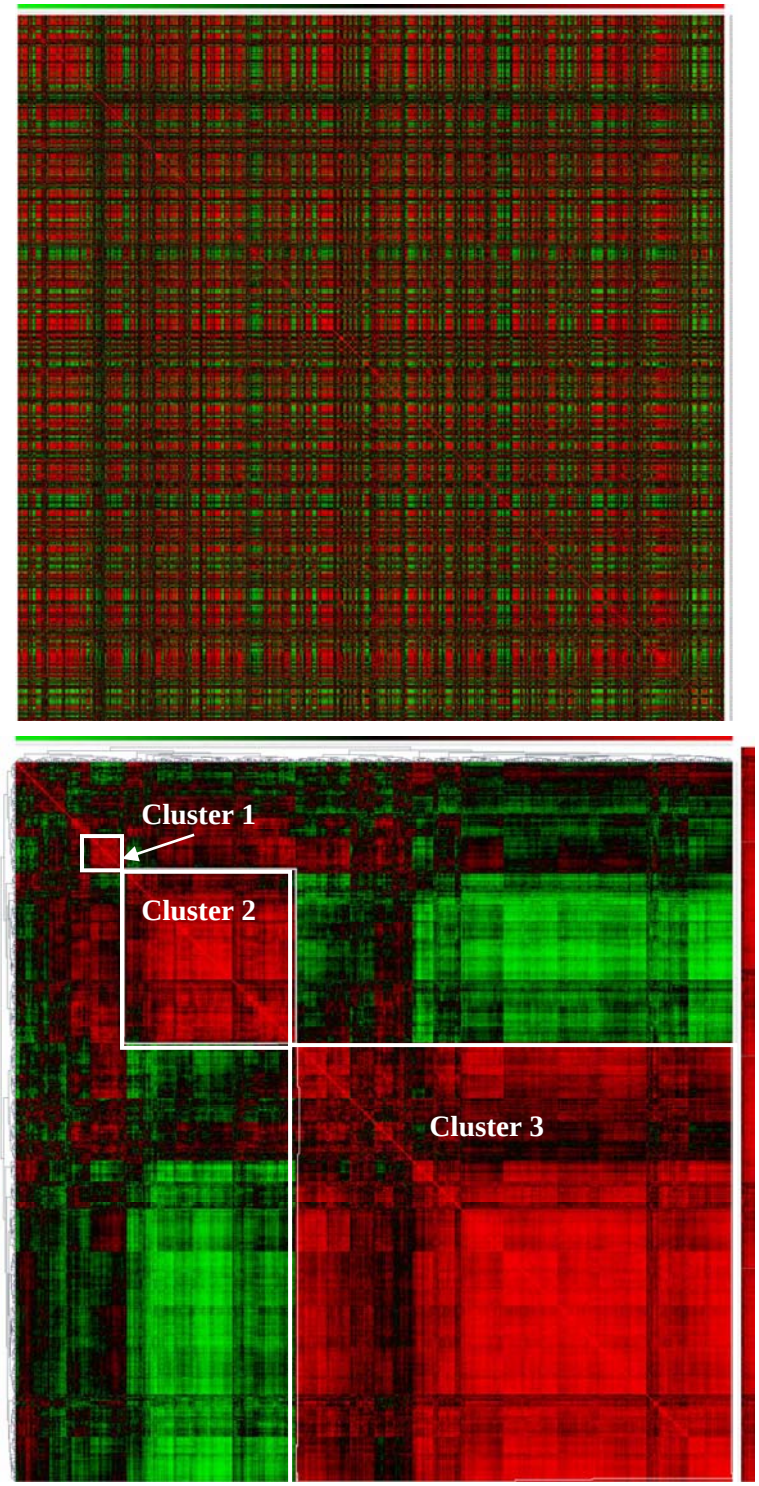


Figure 5-3 Gene coexpression matrix of *E.coli.*

Top: Gene coresponse matrix of *E.coli* genes plotted with Pearson's correlation coefficient (genes arranged in a sequence as appear on chromosome), Red colour indicates Pearson's coefficient of 1 and Green colour indicates the value of -1. Black indicates the value of 0.

Bottom: Clustering analysis, Euclidean distance cluster analysis (MeV software) performed on the top matrix, results in clusters with large number of correlated genes.

To identify genes with similar coexpression, a hierarchical clustering technique with Euclidean distance (complete linkage method) (Eisen *et al.*, 1998) was applied to the correlation matrix using MeV software.

Figure 5-3 (bottom figure) shows three main clusters were observed. Analysis of the clusters-3 (shown with white border squares on the right in the bottom matrix of Figure 5-3) genes shows more than 523 genes positively correlated with each other on the basis Pearson's correlation coefficient. Though it might be possible that all these genes are coexpressed, obtaining correct functionally significant genes from such a large cluster needs further selection criteria. With out any constraint, there is a possibility of associating genes by guilt by association in such a large coexpression data.

It was observed that,

- Functionally irrelevant genes were clustered into the same cluster
- No significant conclusion could be drawn on the genes in the same cluster.
- Only three or four clusters can be identified from such a large coexpression data.

The possible reason for such results may be lack of additional constraints while grouping the genes or due to the similarity of coexpression of more than two genes in the co-response data. Previous reports suggests that applying such traditional clustering may lead to the problem of 'association by-guilt' (Wolfe *et al.*, 2005) where genes in a same cluster significantly differ functionally but are putatively assigned to be of similar function.

5.3.2 **Enzyme subset and coresponse correlations**

To identify functionally similar genes from the co-response data, a new approach was developed based on the structural modelling technique. Genes responsible for reactions in an enzyme subset were considered as a cluster of genes (refer Figure 5.2). The co-response matrices were arranged such that genes in a subset were adjacent one to each others. Each subset was studied for the confidence of the pair wise Pearson's correlation coefficient using the p-

Table 5-2 Enzyme subset classification based on the confidence of gene-coexpression correlation of genes in the subset

Class	Description	Class name
1	Subsets with all the gene-pairs pass confidence test, Complete fit, forming a clique of genes	Complete-fit gene clique
2	Subsets with few gene-pairs pass confidence test from all the gene-pairs	Partial-fit gene clique
3	Subsets with no gene-pairs pass confidence test.	Poor-fit gene clique
4	Subsets with only one gene-pair	--
	4A - Single gene-pair passing confidence test	Single pair-best fit
	4B - Single gene-pair not passing confidence test	Single pair-poor fit
5	Subsets with only one gene	One gene set
6	Subset with no gene assigned	No gene set
7	Reaction set of all dead reactions	Dead subset

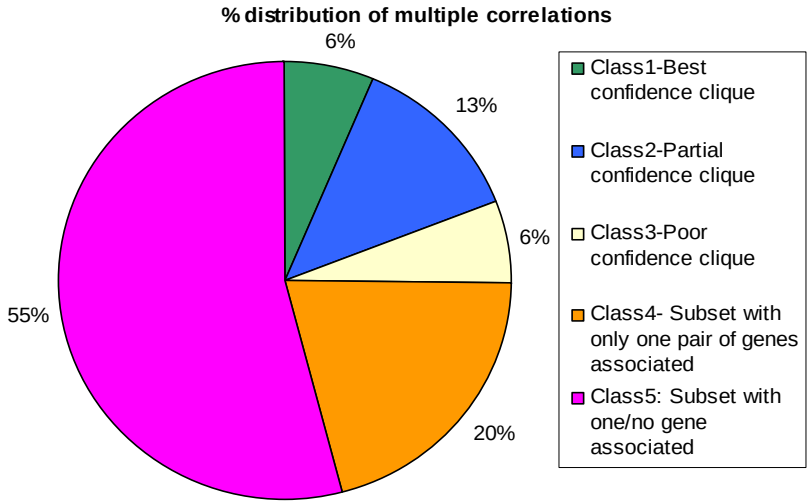


Figure 5-4 Percent distribution of enzyme subset on the basis of confidence of gene-pairs passing BH test.

value. Based on the confidence of the Bonferroni-Holm (BH) test, enzyme subsets were classified into further classes as shown in Table 5-2.

From the above classes, the last two classes (class 6 and 7) were discarded from the further statistical analysis, since no analysis could be performed for such enzyme subsets. The first three classes of enzyme subset were extensively studied to identify significant genes with their coexpression pattern.

Bonferroni-Holm is a very stringent test and therefore the results suffer from possibility of rejecting many significant gene pairs from the dataset. On the other hand, this stringency provides high statistical confidence in the result.

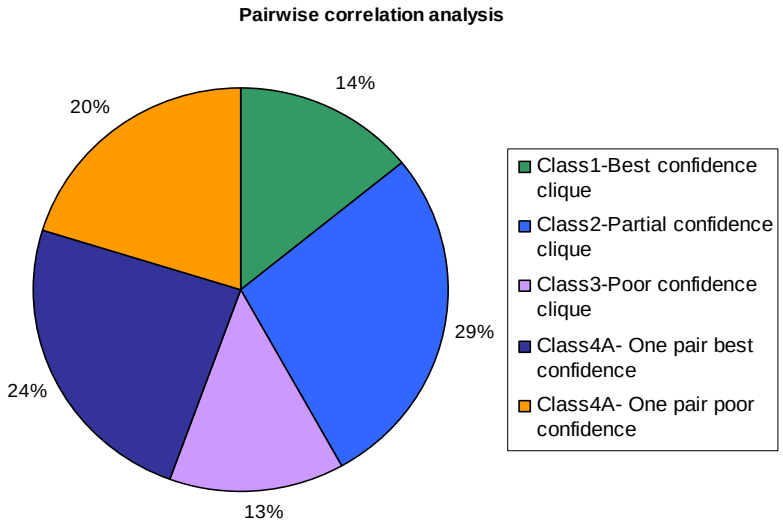


Figure 5-5 Percent distribution of gene pairs in the subset with respect to Bonferroni-Holm (BH).

5.3.3 Discussion on selected enzyme subsets and gene coexpression correlation

Enzyme subsets from Class-1 and Class-2 were further grouped or classified on the basis of the correlation with operon and regulon match as follows:

A. Group 1: Subsets that correlate with known operon

B. Group 2: Subset that may correlate or partially matches with predicted/proposed[45] operon and regulons.

[45] The proposed operons and regulons are the one proposed during the present study on the basis of gene pair confidence with BH test.

Few cases from Group1 and 2 are discussed here.

A. Selected examples from Group 1 subsets:

A-1. Subset-27: This subset which belongs to class-1 consists of 10 reactions which correspond to 8 genes on gene-protein-reaction association.

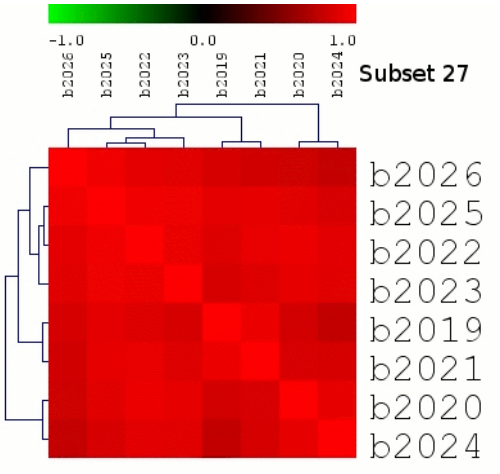


Figure 5-6 Co-response matrix of genes in subset-27

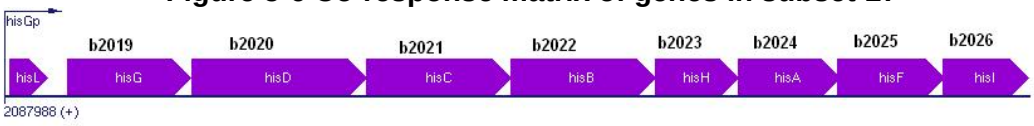


Figure 5-7 Histidine operon in *E.coli*

Table 5-3 Gene-protein-reaction association for subset 27

Gene		Reaction	Protein/Enzyme or function
B2019	hisG	EC-2.4.2.17-ATPPRT	ATP phosphoribosyltransferase
B2020	hisD	EC-1.1.1.23-HISTD	Histidinol dehydrogenase
B2021	hisC	EC-2.6.1.9-HSTPT	Histidinol-phosphate aminotransferase
B2022	hisB	EC-4.2.1.19-IGPDH	Bifunctional enzyme : histidinol phosphatase
B2022	hisB	EC-3.1.3.15-HISTP	Bifunctional enzyme imidazoleglycerolphosphate (IGP) dehydratase
B2023	hisH	IG3PS	Amidotransferase component of imidazole glycerol phosphate (IGP) synthase
B2024	hisA	EC-5.3.1.16-PRMICIi	N-(5'-phospho-L-ribosylformimino)-5-amino-1-(5'-phosphoribosyl)-4-imidazolecarboxamide 636 isomerase
B2025	hisF	IG3PS	Cyclase component of imidazole glycerol phosphate (IGP) synthase
B2026	hisI	EC-3.6.1.31-PRATPP	Bifunctional enzyme: PR-ATP pyrophosphatase
B2026	hisI	EC-3.5.4.19-PRAMPC	Bifunctional enzyme: PR-AMP cyclohydrolase

The lumped reaction for this subset is:
5-Phospho-alpha-D-ribose1-diphosphate + L-Glutamine + ATP + 2 NAD + 3 H_2O → 5-Amino-1-(5-Phospho-D-ribosyl)imidazole-4-carboxamide + 2-Oxoglutarate + 2 NADH + L-Histidine + Pi + 2 PPi + 5 H

This is a best example of perfect match between an operon and a subset. All reactions in a subset are involved in histidine metabolism. Genes encoding these reactions occur as a functional unit, i.e. histidine operon. All the gene pairs show high correlation coefficients and pass both statistical tests.

A-2. Subset 44: This subset belongs to class-2 and consists of 11 reactions associated with 11 genes. Reaction PPTGS do not have any gene assigned yet. This clique is an example of a complex transcription unit. The present analysis suggest that there are two distinct operons of which one is known as part of putative transcription unit 'mra'. The transcription unit shown in Figure 5-9 was putatively assigned (Hara *et al.*, 1997).

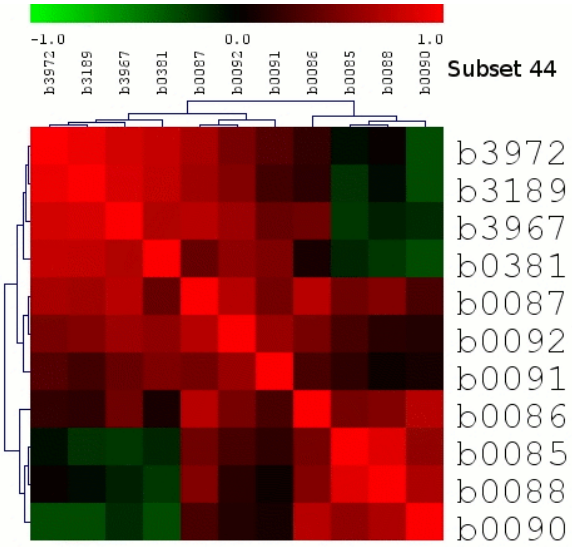


Figure 5-8 Co-response matrix of genes in subset-44

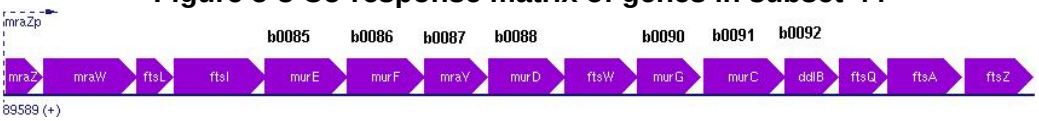


Figure 5-9 A putative transcription unit of mra

Table 5-4 Gene-protein-reaction association for subset 44

Gene		Reaction	Protein or enzyme or function
B0085	murE	EC-6.3.2.13-UAAGDS	UDP-N-acetylmuramoyl-L-alanyl-D-glutamate:meso-diaminopimelate ligase
B0086	murF	EC-6.3.2.15-UGMDDS	UDP-N-acetylmuramoyl-tripeptide:D-alanyl-D-alanine ligase
B0087	mraY	EC-2.7.8.13-PAPPT3	UDP-N-acetylmuramoyl-pentapeptide:undecaprenyl-PO4 phosphatase
B0088	murD	EC-6.3.2.9-UAMAGS	D-glutamic acid adding enzyme; UDP-N-acetylmuramoyl-L-alanine:D-glutamate ligase
B0090	murG	UAGPT3	N-acetylglucosaminyl transferase; UDP-NAc-glucosamine:NAc-muramyl-pentapeptide pyrophosphoryl-undecaprenol NAc-glucosamine transferase
B0091	murC	EC-6.3.2.8-UAMAS	L-alanine adding enzyme; UDP-N-acetylmuramate:L-alanine ligase
B0092	ddlB	EC-6.3.2.4-ALAALAr	D-Alanine:D-alanine ligase
B0381	ddlA	EC-6.3.2.4-ALAALAr	D-Alanine:D-alanine ligase, ADP-forming
B3189	murA	EC-2.5.1.7-UAGCVT	UDP-N-acetylglucosamine enoylpyruvyl transferase; phosphomycin resistance
B3967	murI	EC-5.1.1.3-GLUR	Glutamate racemase; D-glutamate synthesis
B3972	murB	EC-1.1.1.158-UAPGR	UDP-N-acetylglucosaminyl-3-enolpyruvate reductase
NA		PPTGS	Peptidoglycan subunit synthesis

The lumped reaction for this subset is:
2 UDP-N-acetyl-D-glucosamine + L-Alanine + meso-2,6-Diaminoheptanedioate + Phosphoenolpyruvate + L-Glutamate + Undecaprenyl phosphate + 2 D-Alanine + NADPH + 5 ATP → UMP + 6 H + Peptidoglycan subunit of Escherichia coli + UDP + 5 ADP" + 6 Pi + NADP + Undecaprenyl diphosphate

Although several potential promoter sequences in this region of the mra cluster have been reported, Hara *et al.* (1997) concluded that the promoter mraZp may be required for the first nine genes of the cluster to be fully expressed. However, genes located from murG to ftsZ are not exclusively dependent on the mraZp promoter.

A-3. Subset 56 and 112: These subsets belong to class-2. Reactions in both subsets carry isoleucine and valine biosynthesis in *E.coli*.

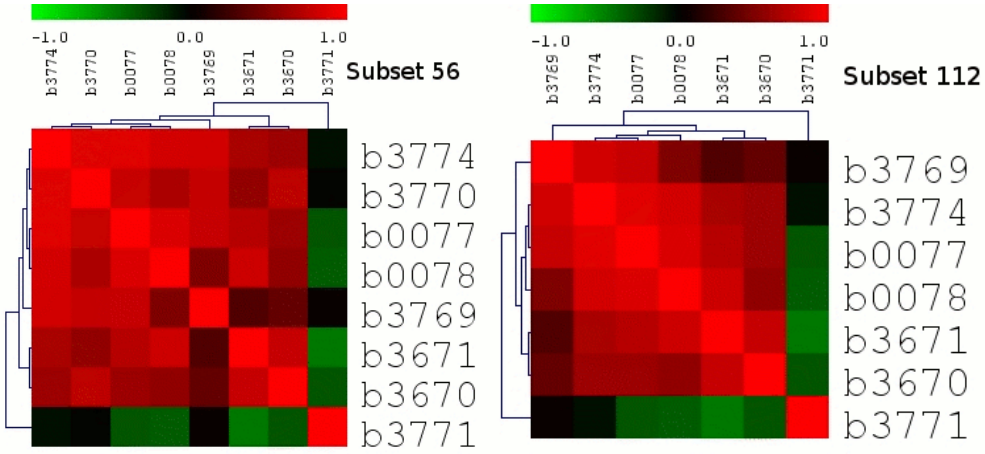


Figure 5-10 Coresponse matrices of subset 56 and 112

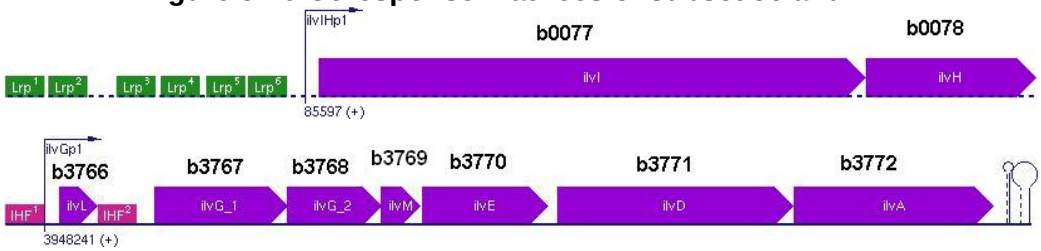


Figure 5-11 ilvIH and ilvLGMEDA operon in *E.coli*

Table 5-5 Gene-protein-reaction association for subset 56

Gene		Reaction	Protein or enzyme or function
b0077	ilvI	ACHBS	Acetohydroxy acid synthase III (AHAS-III); acetolactate synthase III (ALS-III); valine sensitive; large subunit
b0078	ilvH		
b3670	ilvN	ACHBS	Acetohydroxy acid synthase I (AHAS-I); acetolactate synthase I (ALS-I); valine sensitive; small and large subunit
b3671	ilvB		
b3769	ilvM	ACHBS	Acetohydroxy acid synthase II (AHAS-II); acetolactate synthase II (ALS-II); valine insensitive; small subunit
b3770	ilvN	EC-2.6.1.42-ILETA	Branched-chain amino acid aminotransferase
b3771	ilvD	DHAD2	Dihydroxy-acid dehydratase; homodimeric
b3774	ilvC	EC-1.1.1.86-KARA2i	Ketol-acid reductoisomerase
b4488	ilvG_2	ACHBS	Gene not present in the CSB-Golm datasets

Table 5-6 Gene-protein-reaction association for subset 112

Gene		Reaction	Protein or enzyme or function
b0077	ilvI	EC-4.1.3.18-ACLS	Acetohydroxy acid synthase III (AHAS-III); acetolactate synthase III (ALS-III);

b0078	ilvH		valine sensitive; large subunit
b3670	ilvN	EC-4.1.3.18-ACLS	Acetohydroxy acid synthase I (AHAS-I); acetolactate synthase I (ALS-I); valine sensitive; small subunit
b3671	ilvB		
b3769	ilvM	EC-4.1.3.18-ACLS	Acetohydroxy acid synthase II (AHAS-II); acetolactate synthase II (ALS-II); valine insensitive; small subunit
b3771	ilvD	EC-4.2.1.9-DHAD1	Dihydroxy-acid dehydratase; homodimeric
b3774	ilvC	EC-1.1.1.86-KARA1i	Ketol-acid reductoisomerase
b4488	ilvG_2	EC-4.1.3.18-ACLS	Gene not present in the CSB-Golm datasets

The lumped reaction for subset 56 is:

NADPH + Pyruvate + 2 H + L-Glutamate + 2-Oxobutanoate → CO_2 + H_2O + NADP + 2-Oxoglutarate + L-Isoleucine

The lumped reaction for subset 112 is:
NADPH + 2 Pyruvate + 2 H → 3-Methyl-2-oxobutanoate + H_2O + NADP + CO_2

These subsets show a highly complex structure. Gene b0077 and b0078 belong to an operon ilvIH, while b3769, b3770, b3771 and b3772 belong to operon ilvLG_1G_2MEDA. This operon is full of single gene transcription units allowing individual genes to act as one gene operon (Karp, 2006). The negative correlation of gene b3771 suggests this gene might act as a single gene operon in *E.coli*.

A-4. Subset 68: This subset belongs to class-1 category. It consists of 5 reactions with 5 genes associated to these reactions. As shown in Figure 5-13, all the genes in the subset match perfectly with astCADBE operon. This supports the fact that the putative assignment of the genes b1744, b1746 and b1747 is correct as the protein structure of arginine succinyltransferase (b1747) and succinylglutamate desuccinylase (b1744) are yet to be determined.

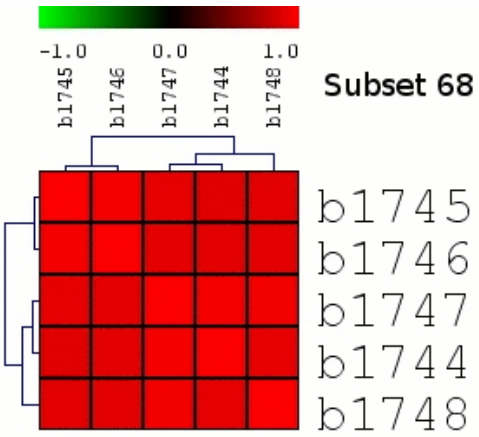


Figure 5-12 Co-response matrix for subset 68

Figure 5-13 Arginine catabolism operon (astCADBE) in *E.coli*

Table 5-7 Gene-protein-reaction association for subset 68

Gene		Reaction	Protein or enzyme or function
b1744	astE	SGDS	Putative succinylglutamate desuccinylase
b1745	astB	EC-2.6.1.69-SADH	Succinylarginine dihydrolase, mutant cannot catabolise arginine
b1746	astD	SGSAD	Putative succinylglutamic semialdehyde dehydrogenase
b1747	astA	EC-2.3.1.109-AST	Putative arginine succinyltransferase
b1748	astC	SOTA	Succinylornithine transaminase, mutant cannot catabolize arginine, overproduction complements argD mutants; carbon starvation protein

The lumped reaction for subset 68 is:
L-Arginine + Succinyl-CoA + 4 H_2O + NAD + 2-Oxoglutarate $\rightarrow$ Succinate + Coenzyme A + H + CO_2 + NADH + 2 NH_3 + 2 L-Glutamate

A-5. Subset 571: This class-2 subset contains only one transport reaction and 6 genes are associated with it.

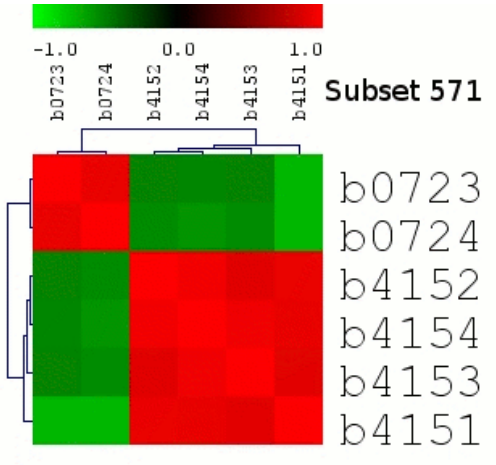


Figure 5-14 Co-response matrix for subset 571

Figure 5-15 frdABCD transcription unit in *E.coli*

Table 5-8 Gene-protein-reaction association for subset 571

Gene		Reaction	Protein or enzyme or function
b0723	sdhA	EC-1.3.99.1-SUCD1i	Succinate dehydrogenase (SQR) flavoprotein subunit; negatively regulated by ryhB RNA as part of indirect positive regulation by Fur
b0724	sdhB	EC-1.3.99.1-SUCD1i	Succinate dehydrogenase (SQR) iron-sulfur protein; negatively regulated by ryhB RNA as part of indirect positive regulation by Fur
b4151	frdD	EC-1.3.99.1-SUCD1i	Fumarate reductase membrane anchor polypeptide
b4152	frdC	EC-1.3.99.1-SUCD1i	Fumarate reductase membrane anchor polypeptide
b4153	frdB	EC-1.3.99.1-SUCD1i	Fumarate reductase iron-sulfur protein subunit
b4154	frdA	EC-1.3.99.1-SUCD1i	Fumarate reductase flavoprotein subunit

The lumped reaction of this subset is:
Succinate + FAD -> $FADH_2$ + Fumarate

The subset is a good example of the wrong Gene-Protein-Reaction association. In the metabolic model, the reaction EC-1.3.99.1-SUCD1i converts Succinate to Fumarate using FAD/FADH2. In *E.coli*, this conversion can occur by two reactions, fumarate reductase (EC 1.3.5.-) and succinate dehydrogenase (EC 1.3.5.1) (Cecchini *et al.*, 1995). In the above case, genes from one enzyme were wrongly assigned to the other enzyme. Gene b0723, b0724, b0721 and b0722 encode for succinate dehydrogenase, while b4151, b4152, b4153 and b4154 encode for fumarate reductase. This wrong assignment was caused due to the EC classification of class-99, the entry 1.3.99.1 of the enzyme was later modified to two separate entries depending on the presence of cofactor in a reaction.

A-6. Subset 91 and 558: These subsets belong to class-2. Subset 91 consists of 4 reactions associated to 10 genes while subset 558 consists of 2 reactions associated to 6 genes. As the ABC transport reactions are encoded by the same protein and hence the same genes, both subsets consist of common genes.

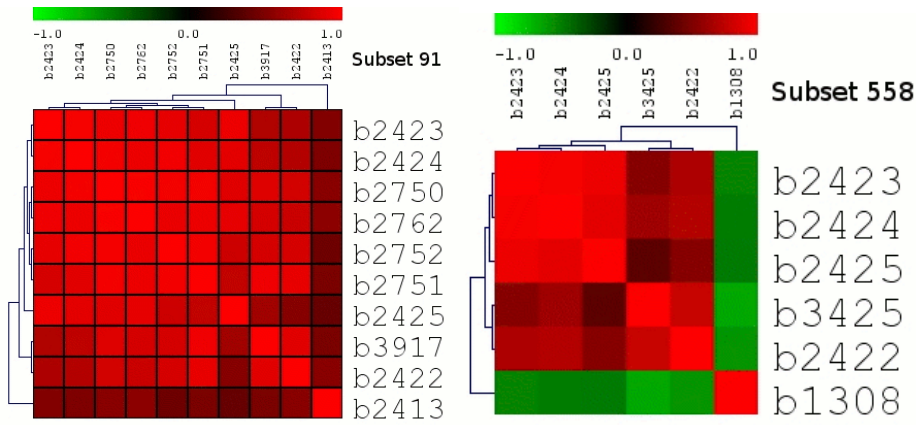


Figure 5-16 Co-response matrices for subset 91 and 558

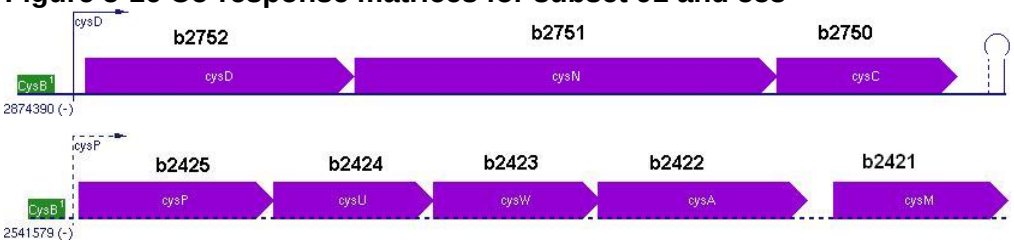


Figure 5-17 cysDNC and cysPUWAM operon in *E.coli*

Table 5-9 Gene-protein-reaction association for subset 91

Gene		Reaction	Protein or enzyme or function
b2413	cysZ	SULabc	ORF upstream of cysK
b2422	cysA	SULabc	Sulfate permease; chromate resistance
b2423	cysW	SULabc	Membrane-bound sulfate transport protein; may also transport molybdate
b2424	cysU	SULabc	Cysteine transport system; may also transport molybdate
b2425	cysP	SULabc	Periplasmic sulfate binding protein
b2750	cysC	EC-2.7.1.25-ADSK	Adenylylsulfate kinase; APS kinase
b2751	cysN	SADT2	ATP sulfurylase (ATP:sulfate adenylyltransferase)
b2752	cysD	SADT2	Sulfate adenylyltransferase
b2762	cysH	EC-1.8.4.8-PAPSR	Phosphoadenylyl sulfate (PAPS) reductase
b3917	Sbp	SULabc	Periplasmic sulfate-binding protein

Table 5-10 Gene-protein-reaction association for subset 558

Gene		Reaction	Protein or enzyme or function
b1308	pspE	EC-2.8.1.1-CYANST	Rhodanase, thiosulfate: cyanide sulfurtransferase; expressed in response to stress as part of psp operon, but also transcribed independently
b2422	cysA	TSULabc	Sulfate permease; chromate resistance
b2423	cysW	TSULabc	Membrane-bound sulfate transport protein; may also transport molybdate
b2424	cysU	TSULabc	Cysteine transport system; may also transport molybdate
b2425	cysP	TSULabc	Periplasmic sulfate binding protein
b3425	glpE	EC-2.8.1.1-CYANST	Thiosulfate:cyanide sulfurtransferase (rhodanase); in glpEGR operon, induced by glycerol

The lumped reaction for subset 91 is:
GTP + Reduced thioredoxin + 2 H_2O + Sulfate_EXTCELL + 3 ATP → PPi + Sulfite + Adenosine 3,5-bisphosphate + Oxidized thioredoxin + GDP + 4 H + 2 ADP + 2 Pi

The lumped reaction for subset 558 is:
H_2O + Thiosulfate_EXTCELL + Cyanide + ATP → Sulfite + Pi + 2 H + Thiocyanate + ADP

The subset 91 shows good correlation with operon cysDNC and partial correlation with transcription unit cysFUWAM which was identified as a potential operon and reported by Sirko *et al.* (1990).

The subset 558 contains two reactions, one of the reactions EC-2.8.1.1 is catalysed by enzyme thiosulfate sulfurtransferase EC-2.8.1.1. This enzyme can be encoded by two different genes, b1308 (pspE) and b3425 (glpE). Interestingly, since only one gene is needed to encode the protein, the negative correlation of b1308 suggests that the other gene b3425 is active for production of the enzyme thiosulfate sulfurtransferase for a given physiological growth condition in *E.coli*.

B. Selected examples of Group 2 subsets:

B-1. Subsets-0: This subset consists of 12 reactions. The gene-protein-reaction association shows that the 12 reactions are associated with the 12 genes as shown below.

This subset belongs to class 2 as it gives a possible regulon on the basis of the coexpression profile from the present study. The only gene showing weak or negative correlation with other genes is b2515 which encodes for the protein 1-hydroxy-2-methyl-2-(E)-butenyl 4-diphosphate synthase.

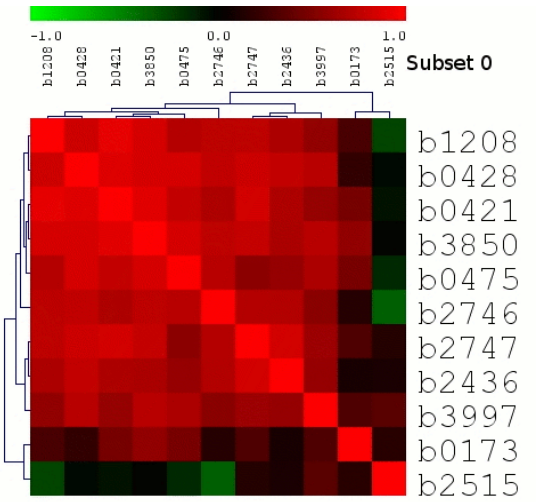


Figure 5-18 Co-response matrix for subset-0

Table 5-11 Gene-protein-reaction association for subset-0

Gene		Reaction	Protein (Enzyme) or function
b0173	dxr (ispC)	DXPRIi	1-deoxy-D-xylulose 5-phosphate reductoisomerase forms 2-C-methyl-D-erythritol 4-phosphate; nonmevalonate (DXP) pathway for terpenoid biosynthesis
b0421	ispA	EC-2.5.1.10-GRTT	Isoprenoid biosynthesis; farnesyl diphosphate synthase
b0421	ispA	EC-2.5.1.1-DMATT	Isoprenoid biosynthesis; farnesyl diphosphate synthase
b0428	cyoE	HEMEOS	Cytochrome o oxidase subunit, protoheme IX farnesyltransferase
b0475	hemH	EC-4.99.1.1-FCLT	Ferrochelatase
b1208	ispE	CDPMEK	4-diphosphocytidyl-2-C-methylerythritol kinase; isopentenyl phosphate kinase; nonmevalonate (DXP) pathway for terpenoid biosynthesis
b2436	hemF	EC-1.3.3.3-CPPPGO	Coproporphyrinogen III oxidase
b2515	ispG	MECDPDH	1-hydroxy-2-methyl-2-(E)-butenyl 4-diphosphate synthase; nonmevalonate (DXP) pathway for terpenoid biosynthesis; [4Fe-4S] protein
b2746	ispF	MECDPS	2C-methyl-D-erythritol 2,4-cyclodiphosphate synthase; nonmevalonate (DXP) pathway for terpenoid biosynthesis; essential gene
b2747	ispD	MEPCT	2-C-methyl-D-erythritol 4-phosphate cytidylyltransferase; nonmevalonate (DXP) pathway for terpenoid biosynthesis; essential gene
b3850	hemG	EC-1.3.3.4-PPPGO	Protoporphyrinogen oxidase activity; neomycin sensitivity
b3997	hemE	EC-4.1.1.37-UPPDC1	Uroporphyrinogen decarboxylase

The lumped reaction for subset 0 is:

6 CTP + 2 Dimethylallyl diphosphate + 20 H + 2 Fe_2 + 4 Isopentenyl diphosphate + 6 (1-deoxy-D-xylulose 5-phosphate) + 2 Uroporphyrinogen III + 6 NADPH + 5 O_2 + 6 ATP $\rightarrow$ 12 PPi + 6 (1-hydroxy-2-methyl-2-[e]-butenyl 4-diphosphate) + 6 CMP + 2 Heme O + 6 NADP + 6 ADP + 12 CO_2 + 14 H_2O

The possible reason for such negative correlations is identified because of the wrong entry in the reaction stoichiometry of MECDPDH reaction. The incomplete stoichiometry of the reaction caused the reaction to appear wrongly into this subset.

B-2. Subset-6: This is a class-2 subset, consists of 6 reactions with 6 corresponding encoding genes.

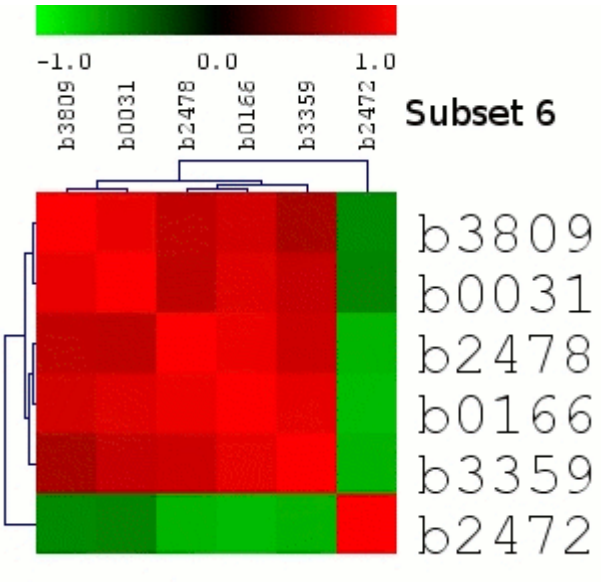


Figure 5-19 Co-response matrix of genes in subset-6

Table 5-12 Gene-protein-reaction association for subset-6

Gene		Reaction	Protein or Function
b0031	dapB	EC-1.3.1.26-DHDPRy	Dihydrodipicolinate reductase
b0166	dapD	EC-2.3.1.117-THDPS	2,3,4,5-tetrahydropyridine-2-carboxylate N-succinyltransferase; mutations suppress growth defects of strains lacking superoxide dismutase
b2472	dapE	EC-3.5.1.18-SDPDS	N-succinyl-diaminopimelate deacylase
b2478	dapA	EC-4.2.1.52-DHDPS	Dihydrodipicolinate synthase
b3359	argD	EC-2.6.1.17-SDPTA	Acetylornithine aminotransferase; succinyldiaminopimelate aminotransferase, PLP-dependent
b3809	dapF	EC-5.1.1.7-DAPE	Diaminopimelate epimerase

The lumped reaction for this subset is:
NADPH + Pyruvate + L-Glutamate + Succinyl-CoA + L-Aspartate 4-semialdehyde → Succinate + Coenzyme A + NADP + 2-Oxoglutarate + meso-2,6-Diaminoheptanedioate

This is an example of complex regulons. All the dap genes clustered together. The reason for negative coexpression of dapE (b2472) is unknown, and suggests that further investigation on the annotation of the dapE gene is needed (Bouvier *et al.*, 1992). Another possibility is that the experimental (microarray) data is incomplete or of a poor quality for this gene.

B-3. Subset 26: This is a class-1 subset and consists of three reactions associated to five genes.

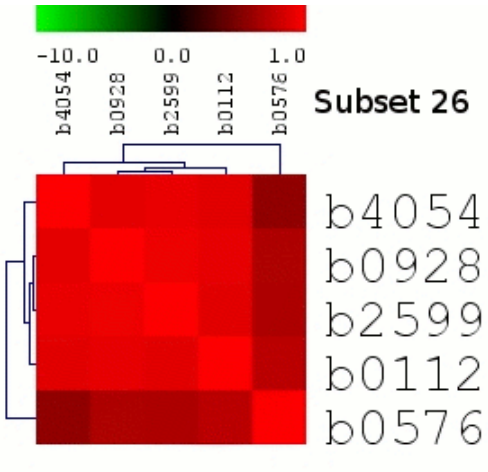


Figure 5-20 Co-response matrix of genes in subset-26

Table 5-13Gene-protein-reaction association for subset-26

Gene		Reaction	Protein/enzyme or function
b0112	aroP	PHEt2r	General aromatic amino acid transport; TyrR regulon
b0576	pheP	PHEt2r	Phenylalanine-specific permease
b0928	aspC	EC-2.6.1.57-PHETA1	Aspartate aminotransferase; kynurenine aminotransferase; glutamine transaminase K
b2599	pheA	EC-4.2.1.51-PPNDH	Phenylalanine synthesis, bifunctional: chorismate mutase (N) and prephenate dehydratase (central); contains Phe-binding regulatory domain (C); FPA resistance
b4054	tyrB	EC-2.6.1.57-PHETA1	Tyrosine aminotransferase; aromatic amino acid aminotransferase; dicarboxylic amino acid aminotransferase; TyrR regulon; homodimeric

The lumped reaction for this subset is:
2 H + L-Glutamate + Prephenate → CO_2 + H_2O + H_EXTCELL + L-Phenylalanine_EXTCELL + 2-Oxoglutarate

Since all the gene pairs show high coexpression profile, these genes must belong to a regulons in the E.coli, though further experimental verification is needed to identify complete regulon structure and assignment of regulons unit.

B-4. Subset 77: This is class-2 subset with 9 reactions and 12 associated genes. In this case b0179, b0180 and b0181 are in an operon whose structure is not yet well understood (Dartigalongue *et al.*, 2001). From the coexpression, we could propose that gene b0182 also belongs to the same operon or shares the same regulation as the other three genes. Gene b2323 is a one gene operon. Other three genes b0096, b0480 and b0915 might appear to share an operon or regulon in *E.coli.*

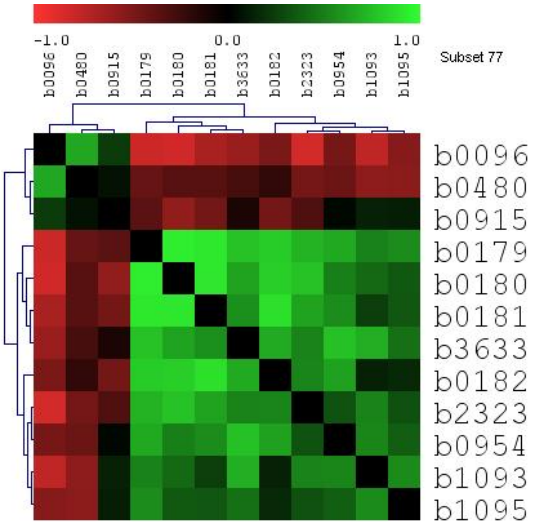


Figure 5-21 Co-response matrix of genes in subset-77

Figure 5-22 lpxPDAfabZ operon in *E.coli*

Table 5-14 Gene-protein-reaction association for subset-77

Gene		Reaction	Protein or enzyme or function
B0096	lpxC	UHGADA	Lipid A biosynthesis, UDP-3-O-(R-3-hydroxymyristoyl)-N-acetylglucosamine deacetylase; zinc metalloamidase; cell envelope and cell separation
B0179	lpxD	U23GAAT	Lipid A biosynthesis, UDP-3-O-(R-3-hydroxymyristoyl)-glucosamine N-acyltransferase
B0180	fabZ	KAS16	3R-hydroxymyristoyl acyl carrier protein (ACP) dehydrase
B0181	lpxA	EC-2.3.1.129-UAGAAT	Lipid A biosynthesis, UDP-N-acetylglucosamine acyltransferase
B0182	lpxB	EC-2.4.1.182-LPADSS	Lipid A disaccharide synthase
B0480	ushA	USHD	UDP-glucose hydrolase and 5'-nucleotidase; bifunctional periplasmic enzyme; monomeric
B0915	lpxK	EC-2.7.1.130-TDSK	Lipid A 4' kinase
B0954	fabA	KAS16	Beta-hydroxydecanoylthioester dehydrase
B1093	fabD	KAS16	3-Ketoacyl-ACP reductase
B1095	fabF	KAS16	Beta-ketoacyl-ACP synthase II; KAS II; homodimeric
B2323	fabB	KAS16	Beta-ketoacyl-ACP synthase I; KAS I; homodimeric
B3633	kdtA	MOAT	3-deoxy-D-manno-octulosonic acid transferase
B3633	kdtA	MOAT2	3-deoxy-D-manno-octulosonate(Kdo)-lipid A transferase

The lumped reaction for subset 77 is:
4 Dodecanoyl-ACP (n-C12:0ACP) + 4 Malonyl-[acyl-carrier protein] + 2 UDP-N-acetyl-D-glucosamine + 2 CMP-3-deoxy-D-manno-octulosonate + 3 H_2O + 4 NADPH + ATP $\rightarrow$ UMP + 2 CMP + UDP + 4 NADP + ADP + 4 CO_2 + 2 Acetate + 8 acyl carrier protein + KDO(2)-lipid IV(A)

B-5. Subset 186: This is class-2 subset with 4 reactions and 11 associated genes. Gene b2091, b2092, b2093, b2094, b2095 and b2096 share the same gat operon in *E.coli* genome. From the coexpression profile, it may possible that genes b2415 and b2416 share a putative operon. In addition, genes b3132 and b3137 may also belong to the same operon.

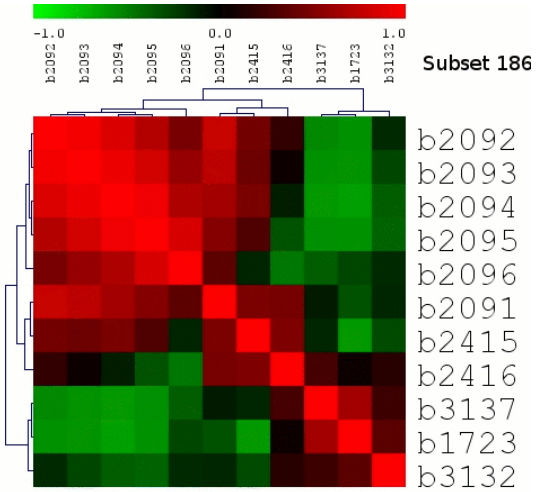


Figure 5-23 Co-response matrix for subset 186

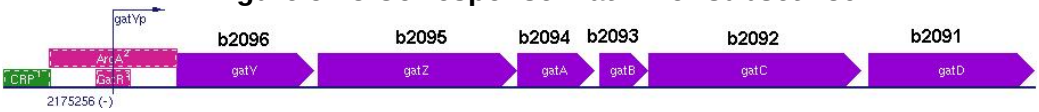


Figure 5-24 gat operon in *E.coli*

Table 5-15Gene-protein-reaction association for subset 186

Gene		Reaction	rotein or enzyme or function
b1723	pfkB	EC-2.7.1.144-PFK_2	Phosphofructokinase, Pfk-2 (PFK II); promoter activation mutation increases expression and suppresses pfkA mutations; tetrameric; allosteric: inhibited by ATP
b2091	gatD	GLTPD	Galactitol-1-phosphate dehydrogenase
b2092	gatC	GALTpts	Galactitol-specific enzyme IIC of PTS
b2093	gatB	GALTpts	Galactitol-specific enzyme IIB of PTS
b2094	gatA	GALTpts	Galactitol-specific enzyme IIA of phosphotransferase system (PTS)
b2095	gatZ	EC-4.1.2.40-TGBPA	Subunit required for full activity and stabilty of GatY tagatose bisphosphate aldolase
b2096	gatY	EC-4.1.2.40-TGBPA	D-Tagatose-1,6-bisphosphate aldolase; requires GatZ subunit for full activity and stability
b2415	ptsH	GALTpts	PTS system histidine phosphocarrier protein Hpr; PP-hexose phosphotransferase
b2416	ptsI	GALTpts	Phosphotransferase system enzyme I
b3132	agaZ	EC-4.1.2.40-TGBPA	Subunit required for full activity and stabilty of the ketose bisphosphate aldolase KbaY
b3137	agaY	EC-4.1.2.40-TGBPA	Ketose 1,6-bisphosphate aldolase; D-tagatose 1,6-bisphosphate aldolase; part of aga cluster for K+ transport; requires KbaZ subunit for full activity and stability; homotetrameric

The lumped reaction for this subset is:
Phosphoenolpyruvate + Galactitol_EXTCELL + NAD + ATP → 2 H + ADP + NADH + Glyceraldehyde 3-phosphate + Dihydroxyacetone phosphate + Pyruvate

B-6. Subset 223: This is a class-2 subset consisting of 10 reactions associated with 10 genes. Most of the reactions of this subset are involved in tyrosine, tryptophan, and phenylalanine metabolism in the model. The weak co-expression of two genes b1692 and b3281 may suggest that these two genes belong to a separate regulon while the rest of the genes share a common regulon in *E.coli*.

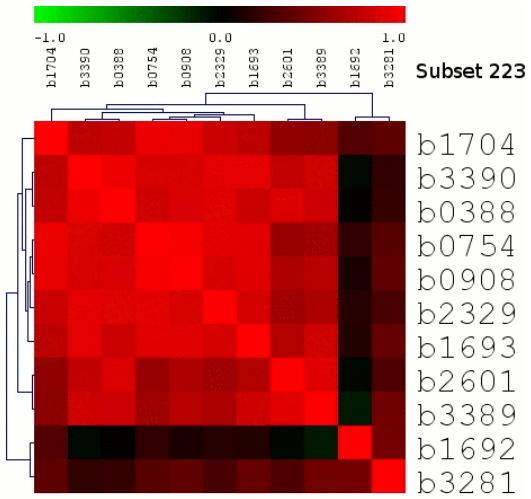


Figure 5-25 Co-response matrix for subset 223

Table 5-16 Gene-protein-reaction association for subset-223

Gene		Reaction	Protein or enzyme or function
B0388	aroL	EC-2.7.1.71-SHKK	Shikimate kinase II
B0754	aroG	EC-4.1.2.15-DDPA	3-deoxy-D-arabino-heptulosonate-7-phosphate (DAHP) synthase; phenylalanine repressible; TyrR regulon
B0908	aroA	EC-2.5.1.19-PSCVT	5-enolpyruvyl shikimate-3-phosphate synthase; ESPS synthase; 3-phosphoshikimate-1-carboxyvinyltransferase
B1692	ydiB	EC-1.1.1.25-SHK3Dr	Quinate/shikimate dehydrogenase, NAD(P)-dependent
B1693	aroD	EC-4.2.1.10-DHQD	3-Dehydroquinate dehydratase
B1704	aroH	EC-4.1.2.15-DDPA	3-deoxy-D-arabino-heptulosonate-7-phosphate (DAHP) synthase; tyrosine repressible
B2329	aroC	EC-4.2.3.5-CHORS	Chorismate synthase
B2601	aroF	EC-4.1.2.15-DDPA	3-deoxy-D-arabino-heptulosonate 7-phosphate (DAH7-P) synthase; tyrosine repressible; TyrR regulons
B3281	aroE	EC-1.1.1.25-SHK3Dr	Shikimate 5-dehydrogenase
B3389	aroB	DHQS	3-Dehydroquinate synthase
B3390	aroK	EC-2.7.1.71-SHKK	Shikimate kinase I; alkali-inducible

The lumped reaction for this subset is:
NADPH + 2 Phosphoenolpyruvate + D-Erythrose 4-phosphate + ATP $\rightarrow$ 4 Pi + NADP+ ADP + Chorismate

5.4 Discussion

5.4.1 Analysis of gene co-expression data

The study of coexpression matrix was used to find the genes with similar functional expression profiles across large number of microarray data. However, such a study needs further analysis which is some what subjective and may not always result in significant gene correlation on the functional level. Therefore additional constraint based coexpression study is needed to avoid the 'guilt by association' assignment of the two genes.

5.4.2 Use of substructure of metabolic network as constraint to study the co-expression profile

The coexpression data clustering on the basis of enzyme subsets shows high correlation amongst the genes in a same subset. More than 25 percent of the subsets show high correlation within the confidence limit of Bonferroni-Holm pair wise correlation tests. This suggests that genes in a subset have a high probability of coexpressional regulation. This finding is important to verify the quality of the functional assignment of the genes in the model organism as discussed in the case study A-5 of selected enzyme subsets of this Chapter. Such a test may highlight the possible wrong functional assignment of genes.

5.5 Conclusion

A genome scale metabolic model can be used as a tool to understand the relationship between the genome and metabolic structure. In the present study, functionally coregulated genes were identified using the structural modelling of 'omics' scale models. Such a model plays a significant role in exploring the complexity of genomic and metabolic interactions. Genes with similar regulatory structure were observed from the metabolic reaction clusters (enzyme subsets) in *E.coli*.

The approach used for studying the relationship between metabolic and genome structure can be further extended to study the transcriptome of the organism by building genomic scale or transcriptome based metabolic models of the organism. Such study not only helps in understanding the system, but also points towards the discrepancies in gene association and gene function assignment. A putative operon/regulon can also be identified from such study; however such assignment will need further investigation and experimental validation.

Chapter 6
Damage analysis of metabolic networks

6.1 Introduction

Recently efforts have been made to study metabolic networks with the help of graph theory approaches (Jeong *et al.*, 2000). In graphical representations of metabolic models, metabolites can be represented by nodes (or vertices) and reactions by edges such as that given by famous Boehringer Biochemical pathways poster[46] (Michal, 1999). One can also represent the metabolic network as bipartite graphs (such as KEGG pathway graphs online, reactions in square node, metabolites in circles, and edges representing their interactions). Such graphical forms of visualisation play an important role in understanding the relationship of reactions and metabolites in the metabolic network, but fail to show the system-wide importance of a metabolite or reaction in complex metabolic networks.

This chapter describes the applications of graph theoretic approaches for the analysis of the genomic scale metabolic networks. The aspects of structural modelling discussed earlier such as the null space and the elementary mode analysis were applied to identify the importance of a reaction by deleting that reaction from the system. Such modified techniques were compared with similar other techniques such as the graph theoretic approach and minimal cut set analysis.

6.2 Graph theory based metabolic network study

6.2.1 Graph theory and metabolic networks

Just as a metabolic network can be recorded as a stoichiometry matrix, a network graph can be represented in a matrix form. Two ways of doing this are:

[46] http://www.expasy.ch/tools/pathways/

• Incidence matrix – In this form, the graph is represented by a matrix of edges by vertices. Matrix columns represent vertices while rows represent edges and elements of the matrix (represented as matrix [edge, vertices]) contain the information about the connectivity of edge to vertex. A stoichiometry matrix is a modified form of the incidence matrix (Zevedei-Oancea & Schuster, 2003).

• Adjacency matrix – In this form, only vertices are considered, the matrix rows and columns both represent vertices of the graph (Aldous *et al.*, 2000). If there is an edge from one vertex 'A' to other vertex 'B', then the element of the matrix [A, B] is 1, otherwise it is 0. In this form, it is easier to find sub graphs, but all other information except the connectedness of the two vertices is lost.

Graphical representation of metabolic network

The graphical metabolic network can be described in three forms as follows:

1. A substrate graph has nodes representing the metabolites and reactions connecting two or more nodes. This is the simplest form of graph representation.
2. A reaction graph has reactions as nodes and metabolites as connections. A connection between two reactions means two reactions involve a common metabolite Such a representation was used by Wagner and Fell (2001).
3. A bipartite graph representation uses a more elaborate way of defining graph, by using two different types of nodes, one for representing metabolites and the other representing reactions or enzymes. In this case, an arc always connects a metabolite to a reaction and vice versa, but never connects two metabolites or reactions directly. Such a representation is much favoured for metabolic maps, since it contains more information about the different components of the system. The bipartite graph representation is also used in Petri nets for modelling metabolic systems (Chen & Freier, 2002; Zevedei-Oancea & Schuster, 2003). KEGG also represents metabolic pathways using a modified form of the bipartite representation.

6.2.2 Network topology analysis based on graph theory

A. General properties of a graph used for network analysis

- **Network path analysis:**

The path length between two nodes is the numbers of edges present between any two nodes in a given network (Aldous *et al.*, 2000). It may be possible to have more than one path length between two nodes in a complex network. Therefore the average of all the path lengths is reported for any two nodes. The most commonly studied other statistical properties of large networks are the degree distribution and network diameter defined as follows.

- **The degree distribution:**

The degree (k) of a node is the number of edges (connections) with the other nodes in the network. The degree distribution, P(k), describes the probability that a node has degree (k) (Newman, 2003). This network property has been used to distinguish between different network models, such as random networks that have a Poisson degree distribution while scale-free networks have a power law degree distribution i.e. $P(k) \propto k^{-\lambda}$, where λ is a positive number (Watts & Strogatz, 1998).

- **Network diameter[47]:**

The smallest number of links that have to be traversed to get from one node to another in a network is called the 'distance' between nodes and a path through the network that achieves this distance is called a shortest path. The average of shortest path lengths over all pairs of nodes in a network is called the network diameter (Watts & Strogatz, 1998).

B. Scale free network

[47] Note that in classical graph theory, the 'diameter' is the maximum length of the shortest path lengths over all pairs of nodes in the network

The scale-free property of certain networks was first reported by Watts and Strogatz (1998). Albert-Laszlo Barabasi and colleagues (Barabasi & Albert, 1999) observed that in World Wide Web the connectedness[48] of network did not have an even distribution around the nodes. They also observed a similar phenomenon in biological and social networks, where a few nodes in the network were more highly connected (also referred to as hubs) than other nodes (Barabasi & Oltvai, 2004). The study shows that probability *P(k)* of a node in the network to be connected with *k* other nodes is proportional to $k^{-\lambda}$ where λ and *k* are constant. Jeong and co-workers (Jeong *et al.*, 2000) also found that for a few other organisms, their metabolic network is a scale free network.

6.3 Damage analysis: concept and basics

Damage analysis in metabolic network is one of the methods to understand the necessity or importance of an enzyme (or reaction) in a living cell. Due to the virtual nature of the metabolic models, it is easy to study the effect of removal of each reaction or metabolite in the system. Thus, the metabolic model system is iteratively interrogated with removal of each enzyme in turn from model to observe functioning of the damaged network. The damage caused by reaction knockout is then identified as a function (e.g. inability to carry flux though the reaction) and damage is calculated for each reaction in the system.

The results obtained from such theoretical simulations are used to predict the effect of gene knockout studies. By performing systematic mutagenesis in a given organism, it is possible to determine the viability (or ability to grow and reproduce on the substrate) (Holme *et al.*, 2003). If an organism is not viable in the absence of the protein (or enzyme) produced by the mutant gene, then the gene (and hence the enzyme) is classified as essential, else it is considered to be non-essential.

[48] Connectedness is the number of edges (or links) connected to a given node.

Using a graph theoretic approach, Lemke *et al.* (2004) performed *in silico* graph theoretic damage analysis on a structural metabolic model. They found that a large fraction of enzymes cause very little damage to the metabolic network, while a few enzymes cause serious or large amounts of damage when removed from the network. On comparison with experimentally generated literature data, they found that genes responsible for encoding reactions (or enzymes) causing a high damage score tend to belong to the essential enzyme (or gene) group.

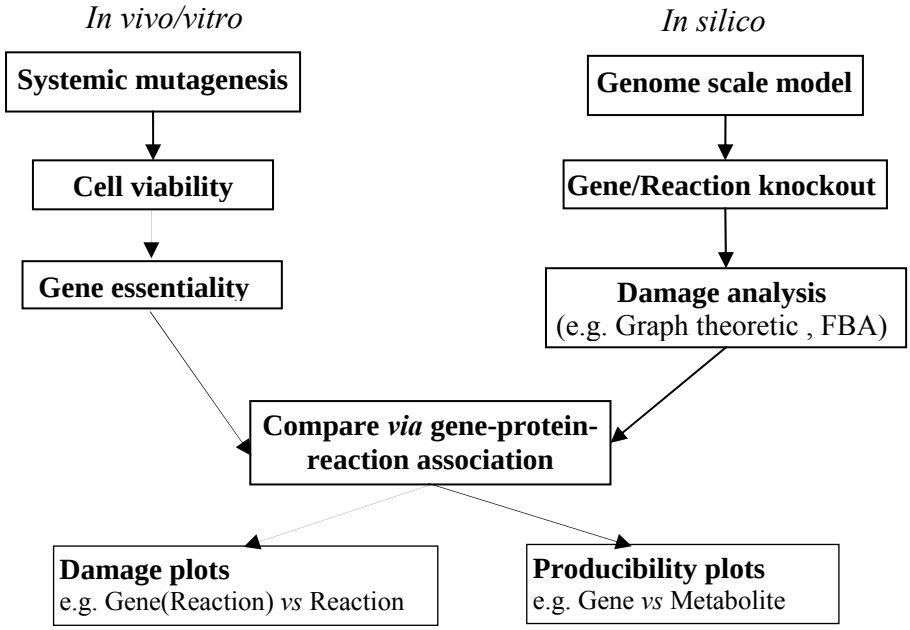


Figure 6-1 Overview of the reaction damage and gene essentiality approach

It is worth mentioning that though their algorithm considers reversibility of the reaction, it does not consider the production of a metabolite by (alternative) other reactions in the system.

6.3.1 Essentiality and damage

The traditional definition of an essential gene is that, when knocked out, it renders the cell unviable. Because essentiality can be determined without knowing the function of a gene (e.g. random transposon mutagenesis (Ross-Macdonald *et al.*, 1999) or gene deletion (Winzeler *et al.*, 1999)), it is a powerful descriptor and starting point for analysis when no other information is available for a particular gene. However, it was observed that non-essential

genes can be found to be synthetically lethal (i.e. cell death occurs when a pair of non-essential genes is deleted simultaneously). It shows that many nonessential genes make significant but small contributions to the fitness of the cell although the effects might not be sufficiently large to be detected by conventional methods.

Essential genes are the genes that are indispensable to sustain cellular life. The functions encoded by essential genes are considered as a foundation of life and therefore are likely to be common for all cells. The essential genes in *E.coli* were reported by Gerdes *et al.* (2003). Another online resource for gene essentiality in *E.coli* is the Profiling of *Escherichia coli* chromosome (PEC[49]) database which gives information on genes and their function and essentiality (Hashimoto *et al.*, 2005).

6.3.2 Other similar approaches to damage analysis

A. FBA based metabolite essentiality

Imielinski *et al.* (2005) used flux balance analysis to approach to study metabolome data. They used flux balance analysis to study the *E.coli* genome scale model for identification of the set of metabolites that get knocked out by damaging a reaction and correlated the metabolite damage with the gene knockout data. They compared producible metabolites between mutant and wild-type strains and produced a metabolite knockout map of *E.coli*. They found that a small fraction of cell membrane, cell wall and quinine metabolites are important for survival of the cell.

B. Minimal cut sets

The concept of minimal cut sets was applied to metabolic networks by Klamt & Gilles (2004). In systems engineering, the concept of minimal cut sets was defined as the set of all the unique combination of component or path that, when they fail, result in failure of the total system. The same concept was

[49] http://shigen.lab.nig.ac.jp/ecoli/pec/index.jsp

implemented in metabolic networks to identify the minimal cut sets as the minimal number of reactions needed to block functioning of a desired reaction. This concept was developed to tackle the problem of the short paths in a graph. It is well known that metabolic network shows presence of few highly connected hubs in the systems (Wagner & Fell, 2001). Minimal cut set provide a minimal set of reactions, which must be blocked, so that the objective reaction will surely also be blocked in a complex system. Recently minimal cut set analysis was used in identification of elementary modes in a metabolic system (Klamt, 2006).

C. Chokepoint analysis

Chokepoint analysis is based on a graph-theoretic approach (Yeh *et al.*, 2004). In this approach, an enzyme is called as a choke-point enzyme if a reaction is known to be catalysed by an enzyme in a metabolic network and,

- if the reaction produces a unique product or metabolite, or
- if the reaction consumes a unique substrate, or
- both of the above, if the reaction consumes and produces a unique metabolite

Yeh *et al.* (2004) performed the choke-point analysis on *Plasmodium falciparum* to identify possible valid targets for drug/vaccine discovery. They identified 216 choke-points in the metabolic network of the organism, of which 87 percent can be potential metabolic drug targets on the basis of comparisons made with host metabolic network.

A similar approach to choke-point analysis was performed on *Salmonella enterica* by Becker *et al.* (2006). They found that the robust metabolism of Salmonella metabolism limits the possibilities of such chokepoint analysis.

A major criticism of this method is that it only accounts for the 'local network analysis' based on the neighbourhood connections. Since this method does not check the essentiality of the metabolite on the systems level, it may be possible that the choke-point enzyme (or metabolites of the reactions catalysed by the

enzyme) may be by-passed by some other alternative route in the metabolic network.

D. Scope of a reaction

This approach includes finding the scope of a reaction or metabolite (Ebenhoh & Handorf, 2004). In this case, a reverse approach is used. After predefining a set of all metabolites and reactions of the metabolic network, one reaction and its substrates are considered as the starting point (called seed). Consecutively, other reactions that can utilise the metabolites available so far are added, and the network is expanded until no further reactions can be added. The set of reactions added after the first starting reaction gives the scope of the first reaction in the network (Ebenhoh *et al.*, 2004).

E. Structural robustness of the network

Wilhelm *et al.* (2004) used the number of elementary modes for finding the structural robustness in the metabolic network. For each reaction (E_i), they defined knockout in terms of ratio z^i/z, where z^i is the number of elementary mode remaining after knockout of E_i, and z is the number of total elementary modes in the unperturbed network. The global robustness (R) of the entire network (i.e. for r number of reactions) is the arithmetic mean of all these numbers and can be given as follows,

$$R = \frac{\sum_{i=1}^{r} z^i}{r.z} \qquad (13)$$

6.4 Damage theory methods

The motivation for the present analysis of metabolic network was to compare different methods such as damage analysis with structural modelling techniques such as null space, flux balance analysis and elementary mode analysis. The graph theoretic methods use local network connectivity while structural methods use global or systems level network properties. It is much easier to compute properties based on the local connectivity in large metabolic networks, whereas

systems analysis methods are much harder to implement for large networks (e.g. elementary mode analysis).

In the present study, three damage analysis methods were compared as shown in Figure 6-2. Graph theoretic damage is based on the local network property while null space damage and elementary mode damage were based on the global or systems level properties of the network. The latter two structural modelling methods may provide better understanding of the cell metabolism. As discussed earlier, the stoichiometry matrix was used to represent metabolic model and the concept of damage analysis was extended to null space and elementary modes analysis.

6.4.1 **Graph theory damage (GTDamage)**

In this method, each enzyme (reaction) was deleted (one at a time) from the metabolic model (i.e. the stoichiometry matrix) to obtain the subsequent damage to the system. The original algorithm methodology was modified to account for the reaction reversibility.

The original damage algorithm proposed by Lemke *et al*. (2004) was designed for a bipartite graph representation of metabolic network and reversible reactions in the network were considered as two separate reactions. In the present study, this algorithm was implemented in Python with following modifications (Poolman, unpublished data[50]);

1) Stoichiometry matrix representation was used for the algorithm

2) Deletion of a metabolite in a reaction was done only if no other reaction produces that metabolite in the metabolic network.

E.g. If a reversible reaction utilises one metabolite and produces two metabolites, removal of that reaction deletes all three metabolites, substrate as well as product metabolites, provided no other reaction is producing those metabolites in the network. In the case of an irreversible reaction, only product

[50] In the present study, the modified graph theoretic damage algorithm was originally proposed by Dr. Mark Poolman. B K B wrote the python implementation, tested and validated using various metabolic models.

metabolites will be deleted from the system, if no other reaction produces the product metabolites of that reaction.

Algorithm for graph theoretic damage (GTDamage):

Pre: OriStoMat: The original stoichiometry matrix of the system
NewStoMat: Copy of the OriStoMat.

1) Choose one reaction (RxDeleted) and delete that reaction from newStoMat.
2) Determine the direction (Reversible/Irreversible) of the RxDeleted
3) If RxDeleted is irreversible:
   a) Delete all the 'product' metabolites of reaction RxDeleted, provided no other reaction produces those product metabolites in the system.
   b) If RxDeleted is reversible, then delete all the metabolites (all 'reactants' and 'products') from RxDeleted, provided no other reaction produce those product metabolites in the system.
4) Determine the set of reactions that utilize any of the deleted metabolites in step 3 and delete all such reactions.
5) Iterate the procedure till no further deletion of reactions is possible from newSM and all the reactions in the newSM are stoichiometrically balanced
6) By comparing OriStoMat and NewStoMat, GTDamage (damage due to RxDeleted) is the set of reactions not present in NewStoMat but present in OriStoMat.
7) Iterate over all the reactions to get GTDamage due to each reaction in the system.

Figure 6-2A gives a schematic representation of GTD analysis on a model system. Consider a simple metabolic reaction system of 11 reactions, with 5 external metabolites. Reactions R1, R3, R5, R7, R8 and R9 are irreversible reactions while other reactions are reversible. Elimination of a (reversible) reaction R2 from the model network deletes R3, R8 and R9 reactions, but G6P is produced by R6 and Rbu5P is produced by R4 and R10. Hence, both metabolites cannot be deleted from the network. Therefore subsequent reactions R1, R6, R4, R5, R10 and R11 also remain undeleted giving the damage set of R2 as R3, R8, and R9. In the case of original damage algorithm proposed by Lemke *et al.* (2004) damage due to R2 would show damage of R3, R4, R5, R6, R7, R8, R9, R10 and R10.

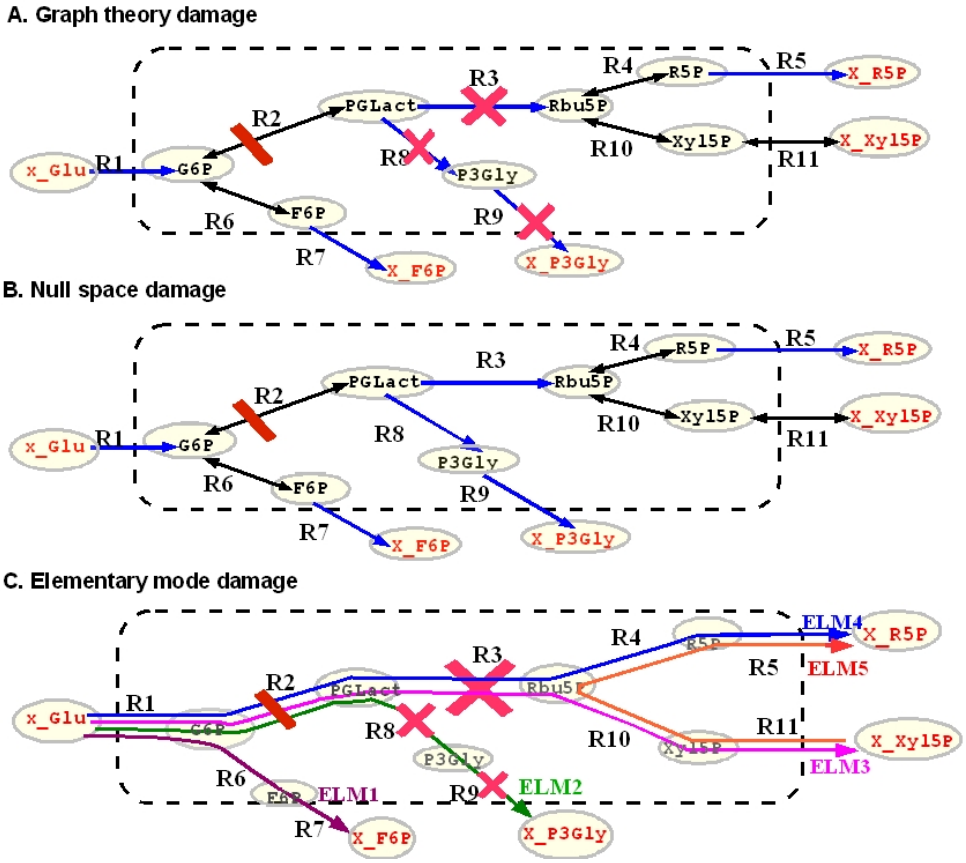


Figure 6-2 Outline of three damage algorithms on a model system

A. Graph theoretic damage; R2 knockout damages three reactions R3, R8 and R9. **B**. Null space damage; R2 knockout does not show any subsequent damage or dead reactions in the system. **C**. Elementary mode damage; R2 knockout causes three modes ELM2, ELM3 and ELM4 to fail; only three reactions R3, R8 and R9 occur only in these failed modes, while other reactions of the system occur in the functioning mode ELM1 and ELM5, therefore, damage due to R2 is R3, R8 and R9.

Table 6-1 Comparison of the three damage methods on the system shown in Figure 6-2

Reac	GTDamage	Score[51]	NSDamage	Score	EMDamage	Score
R1	R1	1	R1	1	R1, R2, R3, R6, R7, R8, R9	7
R2	R2, R3, R4, R5	4	R2	1	R2, R3, R4, R5	4
R3	R3	1	R3	1	R3	1
R4	R4, R5	2	R4, R5	2	R4, R5	2
R5	R5	1	R4, R5	2	R4, R5	2
R6	R6, R7	2	R6, R7	2	R6, R7	2
R7	R7	1	R6, R7	2	R6, R7	2
R8	R4, R5	2	R8, R9	2	R8, R9	2
R9	R9	1	R8, R9	2	R8, R9	2
R10	R10	1	R10, R11	2	R10, R11	2
R11	R11	1	R10, R11	2	R10, R11	2

[51] Damage score for a reaction is calculated as number of reactions damaged by that reaction in the network. Note that minimum damage score for any reaction is one (self damage) while maximum score possible can be total number of reactions of the system.

6.4.2 **Null space damage (NSDamage)**

In this method, the damage to the network is defined as number of dead reactions (reactions carrying no flux at steady state) in the metabolic network. To obtain the null space damage, the original stoichiometry matrix is analysed by null space analysis and all the dead reactions were removed from the stoichiometry matrix. Futile cycles in the network were detected from the null space and all the reactions that only occur in futile cycle were deleted from the original stoichiometry matrix. The reaction under investigation is removed from the stoichiometry matrix and null space analysis was performed to detect the presence of additional dead reactions. The null space damage due to the reaction is the dead reactions in the system.

Algorithm for null space damage (NSDamage):

Pre: OriStoMat: The original stoichiometry matrix of the system
NS: Null space matrix of OriStoMat.

1) Identify 'dead' reactions (if any) from null space and delete them from the originalStoMat. Copy OriStoMat to NewStoMat.
2) Choose one reaction (RxDeleted) and delete that reaction from newStoMat.
3) Calculate null space matrix (NS) for NewStoMat
4) Determine the dead reactions in NS and delete them from NewStoMat.
5) Determine reactions that only occur in futile cycle and delete them from NewStoMat.
6) Iterate over procedure 3, 4 and 5 until no further dead reaction exists in the null space.
7) By comparing OriStoMat and NewStoMat, NSDamage (damage due to RxDeleted) is the set of reactions not present in NewStoMat but present in OriStoMat.
8) Iterate over all the reactions to get NSDamage due to each reaction in the system

As shown in Figure 6-2 B, the null space damage due to R2 is only the reaction itself. This signifies that after removal of the R2 from the system, as per null space damage, no other reaction becomes dead (i.e. all other reactions of the system still carry flux through them).

6.4.3 **Elementary mode damage (EMDamage)**

In this damage analysis method, the damage to the network was defined as the number of reactions explicitly sharing the common elementary modes (EM) or routes that will also be eliminated from the network.

The elementary modes of the metabolic network were computed. Futile cycle modes were then eliminated from the set of elementary modes of the network. For each reaction in consideration, the above set of elementary modes was then divided in to two sets:

a) EM_{dep} modes that uses the reaction.

b) EM_{indep} modes that do not involve the reaction.

The damage for a reaction in consideration by elementary mode analysis is then defined as those reactions that are only present explicitly in EM_{dep} modes and not used by any modes in EM_{indep}.

The elementary mode damage concept is based on the idea of knocking out the metabolic routes of the system. The metabolic routes or elementary routes are the systems level property of a metabolic network; therefore they account for the global properties of the whole system. As mentioned earlier, enzyme subsets or reaction sets form the building blocks of the elementary modes in a metabolic network. Therefore the damage observed by this algorithm is actually a combination of one or more enzyme subsets (including the subset in which the reaction in consideration belongs to) in the metabolic network.

Algorithm for elementary mode damage (EMDamage):

Pre: OriStoMat: The original stoichiometry matrix of the system
EM: elementary of modes for the originalStoMat
NewStoMat: Copy of originalStoMat

1) Calculate elementary modes (EM) for NewStoMat.
2) Remove all the futile modes of the system from the EM.
3) Take one reaction (RxDam) from NewStoMat and determine the set of all the elementary modes (EM_{dep}) passing through that reaction.
4) Determine the set of reactions ($ReacEM_{dep}$) involved in EM_{dep} (i.e. set of all the reactions from all the modes of EM_{dep}).
5) Determine the set of elementary modes (EM_{indep}) not passing through the reaction (RxDam).

6) Determine the set of reactions ($ReacEM_{dep}$) involved in EM_{indep} (i.e. set of all the reactions from all the modes in EM_{indep}).
7) EMDamage is the complement of the two sets $ReacEM_{indep}$ and $ReacEM_{indep}$ (i.e. set of reactions in the only in $ReacEM_{indep}$ but not in the set $ReacEM_{indep}$).
8) Iterate over step 3 to 7 by taking one reaction at a time to get EMDamage score for each reaction in the system.

Figure 6-2-C gives a schematic representation of the elementary mode damage algorithm. To obtain the EMDamage score for R2, first the elementary modes are calculated. The model shows five elementary modes of which three modes (EM_{dep}), ELM2, ELM3 and ELM4 pass through reaction R2 while ELM1 and ELM5 do not pass through R2 (EM_{indep}). The failed modes on deletion of R2 will be ELM2, ELM3 and ELM4 (i.e. EM_{dep}). Reactions of EM_{dep} are R1, R3, R4, R5, R8, R9, R10 and R11 (i.e. $ReacEM_{dep}$). However, R1, R40 R5, R10 and R11 also carry flux through two other elementary modes; ELM5 and ELM1. As system can still carry flux through all these reactions independent of the EM_{dep} modes, these reactions will not be damaged by R2 knockout. Reactions only in $ReacEM_{dep}$ but not in $ReacEM_{indep}$ are R3, R8 and R9 which form the damage set. The process can be repeated for each reaction in the system to obtain elementary mode damage as shown in Table 6-1.

6.5 Metabolic models used for of damage analysis study

Initially small metabolic models with less than 100 reactions such as Calvin cycle model (Poolman *et al.*, 2003), Lactic acid production model (Poolman *et al.*, 2004b) and clavulanic acid synthesis model (Bonde *et al.*, unpublished data) were used.

Due to the problems of computation calculability of elementary modes large models, only GTDamage and NSDamage analysis were performed for *E.coli*-1 and *E.coli*-3 models.

The essential and non essential gene data obtained from the PEC database was used in the present study to correlate the damage due to reaction deletion. A reaction is considered as essential if one or more essential gene encodes for the enzyme responsible for carrying the reaction. Similarly a metabolite is considered essential if it is only produced by an essential reaction in metabolic network. The synthetic lethality of the two nonessential genes was ignored for the present study.

Table 6-2 Comparison between three damage analysis methods

	Graph Theoretic damage	**Null space damage**	**Elementary mode damage**
Reaction Damage defined as	Inability to carry reaction due to non availability of (substrate) metabolite	No flux carrying reaction	Reactions which explicitly use failure elementary modes passing through damaged reaction
Use of network/system property	Uses local connectivity criteria	Uses system wide solution space (null space)	Uses system wide flux (elementary routes) of the system
Network/system size	Very large (almost any) size of network	Moderate size network	Small size networks (~40-100 reactions)
Time of computation	Quick	Slow	Slow
Known problem	Ignores system wide property	Ignores reaction reversibility criteria	--
Computational limitation	--	--	Combinatorial explosion of the elementary modes in a large system in limits the computation

6.6 Results

6.6.1 Damage analysis

Three damage algorithms GTD, NSD and EMD damage analysis were tested for various small size metabolic model systems. To compare the results obtained by these methods with other similar published methods, various well reported models were used.

A comparison in terms of various properties the three damage methods is shown in Table 6-2.

6.6.2 **Minimal cut sets and damage analysis**

To compare between minimal cut sets and the damage algorithms, the same model system shown in Figure 6-3 was used from Klamt & Gilles (2004).

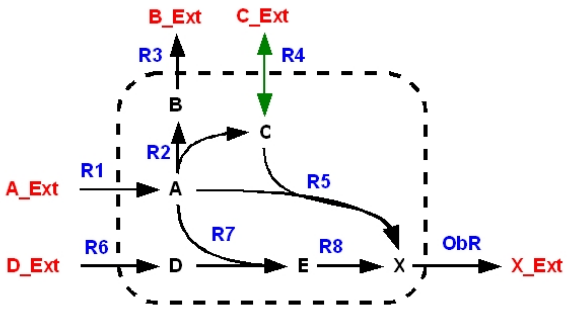


Figure 6-3 Model system for comparing the minimal cut sets and damage (Reproduced from (Klamt & Gilles, 2004)

Figure 6-3 is a model system with only one reversible reaction (R4) and all other reactions (R1, R2, R3, R5, R6, R7, R8 and ObR) of this system are irreversible. External metabolites are A_Ext, B_Ext, C_Ext, D_Ext and X_Ext. Reaction ObR is the objective reaction which produces external metabolite X_Ext. Elementary mode analysis shows that there are 4 elementary modes present in the system of which three pass through reaction ObR and produce X_Ext. The minimal cut sets (MCS) analysis for blocking objective reaction (ObR) performed by Klamt and Gilles (2004) shows that 10 MCSs are possible. Of these 10 cut sets, R5 and R8 is the minimal cut set for blocking the objective reaction ObR. The results from three damage analyses performed on the same model are shown in Figure 6-4, where a red block in the matrix indicates that removal of a reaction (i.e. column name) knock-out reaction of the corresponding row in the matrix.

Using GTDamage analysis, it suggests R1 will surely knockout the 'ObR' reaction, while EMDamage suggests that R1 will knockout all the reaction fluxes in the system

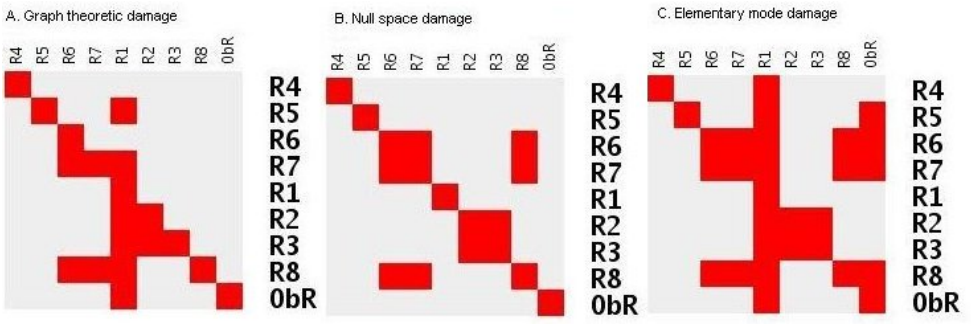


Figure 6-4 Dot-plot representation of GTDamage, NSDamage and EMDamage on a model system shown in Figure 6-3

A. Graph theoretic damage, B. Null space damage, C. Elementary mode damage. Red block indicates that removal of a reaction (as column name) from metabolic model removes (knockouts) corresponding reaction as row names), while square indicates no effect of reaction knock-out.

In NSDamage, reaction R1 does not knock-out any other reaction of the system. On further investigation, it was observed that null space damage suffers from the problem of reaction reversibility[52]. In this case, the system considers that production of metabolite 'A' is possible from the R2 and R3 as all the reactions of the system are treated as reversible. The problem of the null space disregarding the reaction reversibility is obviously not present in elementary mode damage as elementary modes do (strictly) account for the reaction reversibility criteria in the system.

6.6.3 **Damage analysis on Lactic acid synthesis model**

The lactic acid production model developed by Poolman *et al.* (2004b) was analysed for various damages. Figure 6-5 gives the metabolic network of the system while Table 6-3 shows the result of knockout of each reaction with the three damage algorithms.

Comparison of the null space and elementary mode damage shows that they appear to be very similar for most of the reactions of this system. Reactions AcEx, CitEx, CO2Ex, PyrCase and PyrCase show a set of four reactions in NSDamage in addition to self damage. This set of four reactions, CitLyase, TCACycle1, ATPase and AcCoASynth appears to be a false internal cycle,

[52] It is worth to mention that the problem of ignoring reaction reversibility is well known in null space based analysis of the metabolic system. See Section 2.2.5 for further details.

which appear in the null space due to the problem of null space ignoring the reaction reversibility and is wrongly accounted as damage by these reactions.

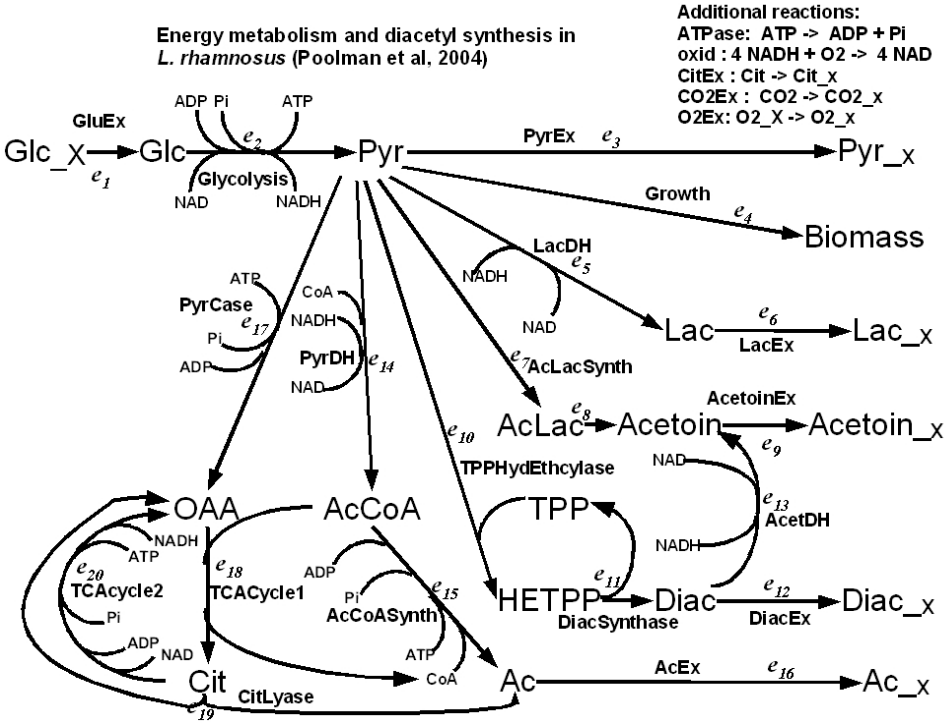


Figure 6-5 Metabolic model of energy and diacetyl synthesis in *L. rhamnosus* reproduced from Poolman *et al.* (2004b)
In this figure the reaction e1, e3, e5, e6, e8, e10, e11, e13, e15, e16, e17 and CitEx should be considered as reversible while rest all other reactions are irreversible.

Table 6-3 Comparison of damage analysis on the lactic acid production model

Reaction	Graph-theoretic	Null space	Elementary mode
AcCoASynth	AcCoASynth	AcCoASynth	AcCoASynth
AcetDH	AcetDH	AcetDH	AcetDH
AcetoinEx	AcetoinEx	AcetoinEx	AcetoinEx
AcEx	AcEx	AcEx, CitLyase, TCACycle1, ATPase, AcCoASynth	AcEx
AcLacDeCase	AcLacDeCase	AcLacDeCase, AcLacSynth	AcLacDeCase, AcLacSynth
AcLacSynth	AcLacSynth	AcLacDeCase, AcLacSynth	AcLacDeCase, AcLacSynth
ATPase	ATPase	ATPase	ATPase
CitEx	CitEx	CitEx, PyrCase, TCACycle1, ATPase, AcCoASynth, CitLyase	CitEx, PyrCase
CitLyase	CitLyase	CitLyase	CitLyase
CO2Ex	CO2Ex	CO2Ex, CitLyase, TCACycle1, ATPase, AcCoASynth	CO2Ex
DiacEx	DiacEx	DiacEx	DiacEx
DiacSynthase	DiacSynthase	TPPHydEthylase, DiacSynthase	TPPHydEthylase, DiacSynthase
GlcEx	Glycolysis, GlcEx	Glycolysis, GlcEx	Glycolysis, GlcEx

Glycolysis	Glycolysis	Glycolysis, GlcEx	Glycolysis, GlcEx
Growth	Growth	Growth	Growth
LacDH	LacDH	LacDH, LacEx	LacDH, LacEx
LacEx	LacEx	LacDH, LacEx	LacDH, LacEx
O2Ex	Oxid, O2Ex	Oxid, O2Ex	Oxid, O2Ex
Oxid	Oxid	Oxid, O2Ex	Oxid, O2Ex
PyrCase	PyrCase	CitEx, PyrCase, TCACycle1, ATPase, AcCoASynth, CitLyase	CitEx, PyrCase
PyrDH	PyrDH	PyrDH	PyrDH
PyrEx	PyrEx	PyrEx, CitLyase, TCACycle1, ATPase, AcCoASynth	PyrEx
TCACycle1	TCACycle1	TCACycle1	TCACycle1
TCACycle2	TCACycle2	TCACycle2	TCACycle2
TPPHydEthylase	TPPHydEthylase	TPPHydEthylase, DiacSynthase	TPPHydEthylase, DiacSynthase

6.6.4 **Damage analysis for the Calvin cycle model**

The Calvin cycle model was taken from Poolman et al. (2004a) and is shown in Figure 6-6. The GTDamage in the system shows all reactions of the system have unique damage score of one (i.e. due to the self damage). In the case of NSDamage, as set of 8 reactions appear as a common set for all the eight reactions in that set (Ru5Pk, X5Piso, Rubisco, TKL2, TKL1, R5Piso, Ald2, SBPase) which is the enzyme subset present in the system. The lumped reaction for this subset is:

$$F6P + 2\ GAP + DHAP + ATP + CO_2 \rightarrow X5P + 2\ PGA + Ru5P + ADP + Pi$$

This subset is retained in the EMDamage. In EMDamage, 13 reactions of the system cause complete damage (all the reactions damaged) in the system as seen from Table 6-4.

An internal cycle present in the system include reactions StSynth, LightReact and StPase. However, null space fails to identify this candidate internal cycle.

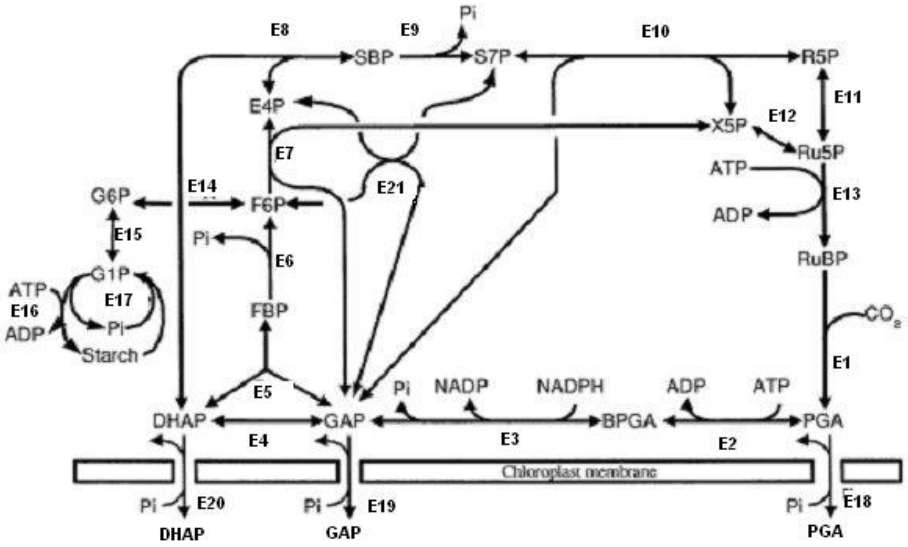


Figure 6-6 Calvin cycle metabolic model reproduced from Poolman *et al.* (2004a)

Reversible Reactions: Rubisco(E1), PGK (E2), G3PDH (E3), TPI (E4), Ald (E5), TKL1 (E7), Ald2 (E8), SBPase (E9), TKL2 (E10), R5Piso (E11), X5Piso (E12), Ru5Pk (E13), PGI (E14), PGM (E15). Irreversible reactions: FBPase (E6), StSynth (E16), StPase(E17), TPT_PGA(E18), TPT_GAP(E19), TPT_DHAP(E20) and additional reaction not shown in system: LightReact :ADP + Pi -> ATP

Table 6-4 Comparison of three damage analyses in the Calvin cycle model

reaction	GTD	NSD	EMD
Ru5Pk	Ru5Pk	Ru5Pk, X5Piso, Rubisco, TKL2, TKL1, R5Piso, Ald2, SBPase	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase
Ald1	Ald1, FBPase	Ald1, FBPase	StSynth, Ald1, FBPase
Ald2	Ald2	Ru5Pk, X5Piso, Rubisco, TKL2, TKL1, R5Piso, Ald2, SBPase	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase
FBPase	FBPase	Ald1, FBPase	StSynth, Ald1, FBPase
G3Pdh	G3Pdh	G3Pdh, PGK	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase
LightReact	LightReact	LightReact	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase
PGI	PGI	PGM, PGI	PGM, StSynth, PGI, StPase
PGK	PGK	G3Pdh, PGK	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase
PGM	PGM	PGM, PGI	PGM, StSynth, PGI, StPase
R5Piso	R5Piso	Ru5Pk, X5Piso, Rubisco, TKL2, TKL1, R5Piso, Ald2, SBPase	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase

Rubisco	Rubisco	Ru5Pk, X5Piso, Rubisco, TKL2, TKL1, R5Piso, Ald2, SBPase	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase
SBPase	SBPase	Ru5Pk, X5Piso, Rubisco, TKL2, TKL1, R5Piso, Ald2, SBPase	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase
StPase	StPase	StPase	StPase
StSynth	StSynth	StSynth	StSynth
TKL1	TKL1	Ru5Pk, X5Piso, Rubisco, TKL2, TKL1, R5Piso, Ald2, SBPase	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase
TKL2	TKL2	Ru5Pk, X5Piso, Rubisco, TKL2, TKL1, R5Piso, Ald2, SBPase	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase
TPI	TPI	TPI	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase
TPT_DHAP	TPT_DHAP	TPT_DHAP	TPT_DHAP
TPT_GAP	TPT_GAP	TPT_GAP	TPT_GAP
TPT_PGA	TPT_PGA	TPT_PGA	TPT_PGA
X5Piso	X5Piso	Ru5Pk, X5Piso, Rubisco, TKL2, TKL1, R5Piso, Ald2, SBPase	Ru5Pk, PGM, TPT_PGA, X5Piso, Rubisco, StSynth, G3Pdh, TKL2, PGK, Ald1, PGI, TPT_DHAP, FBPase, TKL1, TPI, R5Piso, LightReact, Ald2, SBPase, TPT_GAP, StPase

6.6.5 **Damage analysis for Clavulanic acid synthesis model**

The metabolic model for clavulanic acid production in *Streptomyces clavuligenis* (Bonde *et al.*, unpublished data) was used to study the different damage measures. Figure 6-8 gives the details of the metabolic system, while the model specification is included on the supplementary material in CDROM (refer Appendix D for more details).

Figure 6-7 gives the dot plot of the GTDamage. A red block in the matrix indicates that removal of a reaction (i.e. column name) removes (or knock-out) corresponding other reactions (row names) in the matrix. As expected, most of the reactions in the network do not show any significant damage to other reactions in the system by GTDamage.

Figure 6-9 shows the NSDamage of the same system. The NSDamage dot plot is diagonally symmetrical. This is due to the fact that NSDamage algorithm gives the enzyme subset as a damage set. Therefore, each reaction in a subset,

on its knockout, shows a damage of all the reactions present in the enzyme subset. (i.e. reactions of the same enzyme subsets on give same damage in the NSDamage). In addition, null space fails to detect all the futile cycles in the system.

Figure 6-10 shows the dot plot for EMDamage of the system. This plot shows reactions which cause a significant number of (reaction) damage in the system. It is worth to mention that the NSDamage matrix is similar to EMDamage matrix of the system. This suggests a possibility of similarity in the NSDamage and the EMDamage for larger system. However, a valid mathematical proof will be needed to reach to such a conclusion.

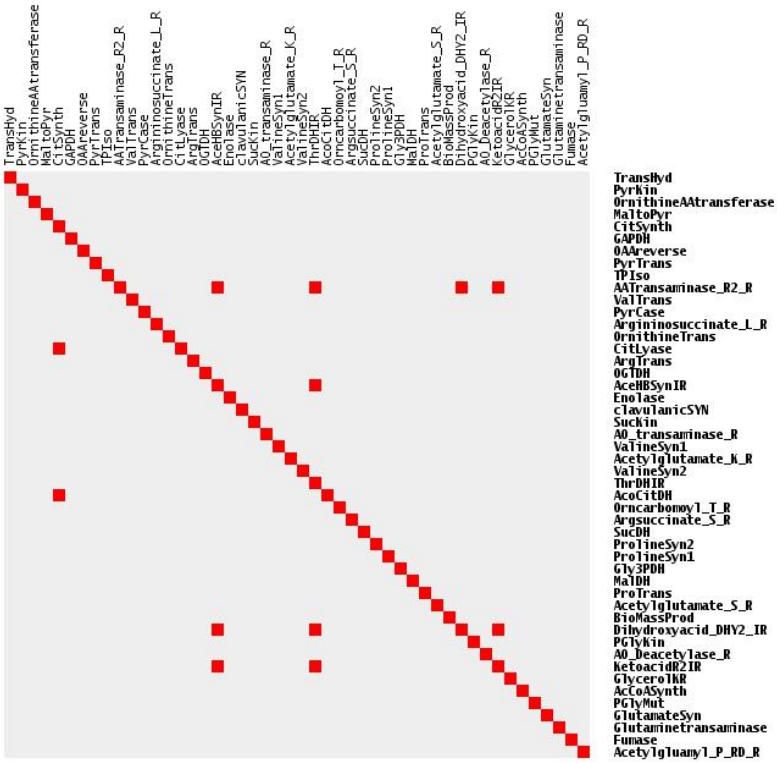


Figure 6-7 Dot-plot of graph theoretic damage (GTD) for the clavulanic acid synthesis model.

A red block in the matrix indicates that removal of a reaction (column name) removes (knock-out) subsequent other reactions (row names) in the matrix

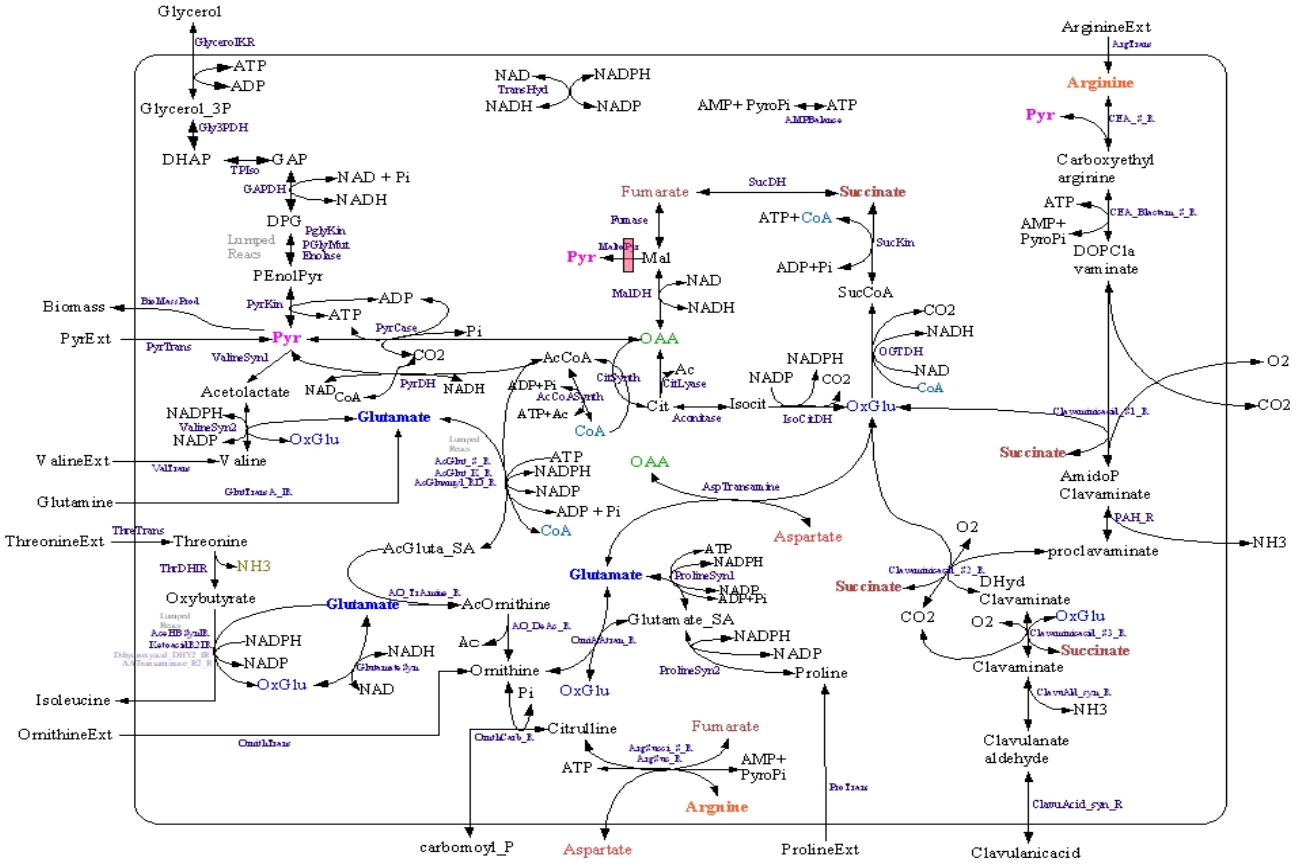


Figure 6-8 The clavulanic acid synthesis metabolic model

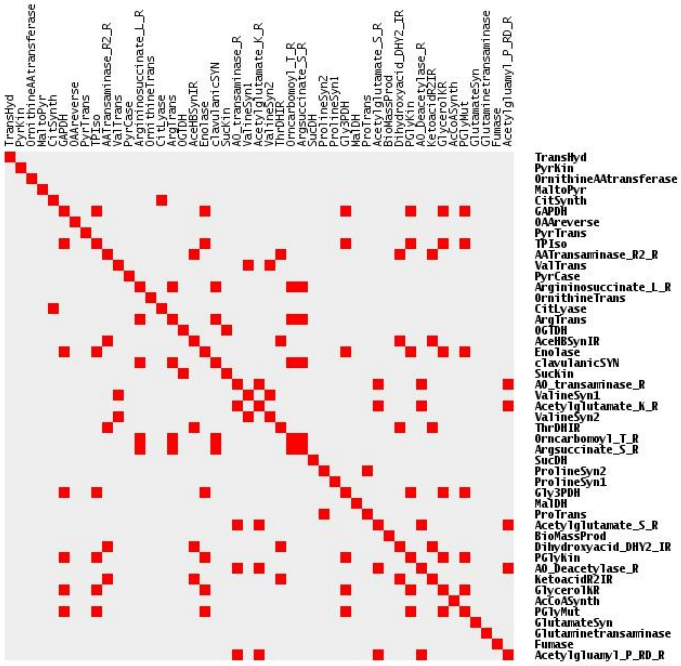


Figure 6-9 Dot-plot of null space damage (NSD) for the clavulanic acid synthesis model.

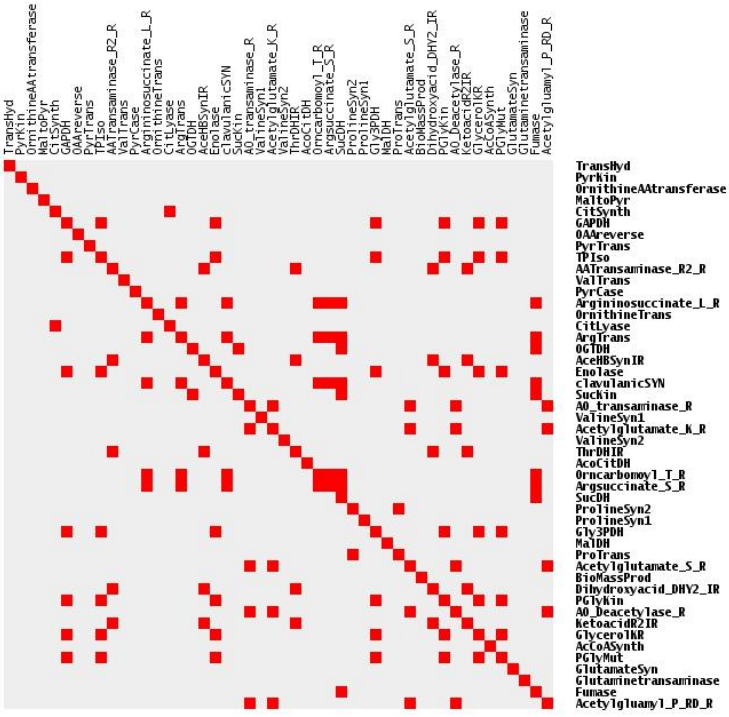


Figure 6-10 Dot-plot of elementary mode damage for the clavulanic acid synthesis model.

A red block in the matrix indicates that removal of a reaction (column name) removes (knock-out) subsequent other reactions (row names) in the matrix

6.6.6 **Comparison of the three damage algorithms for small metabolic networks**

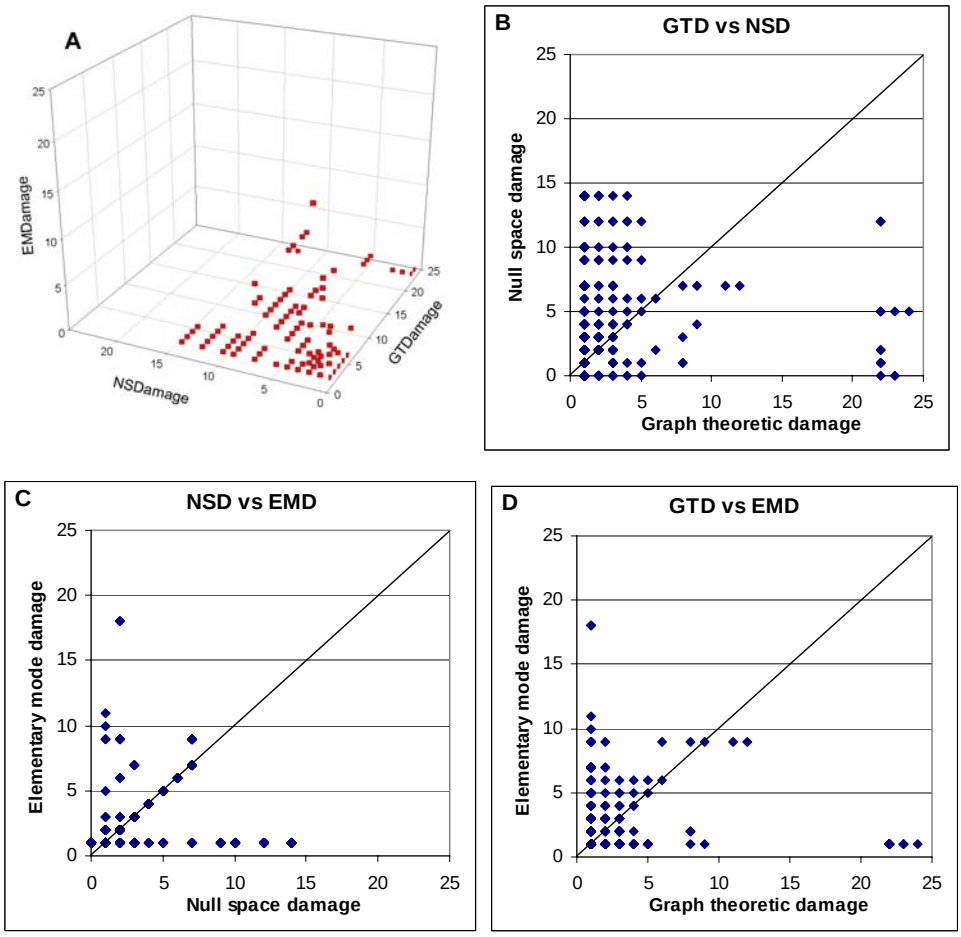
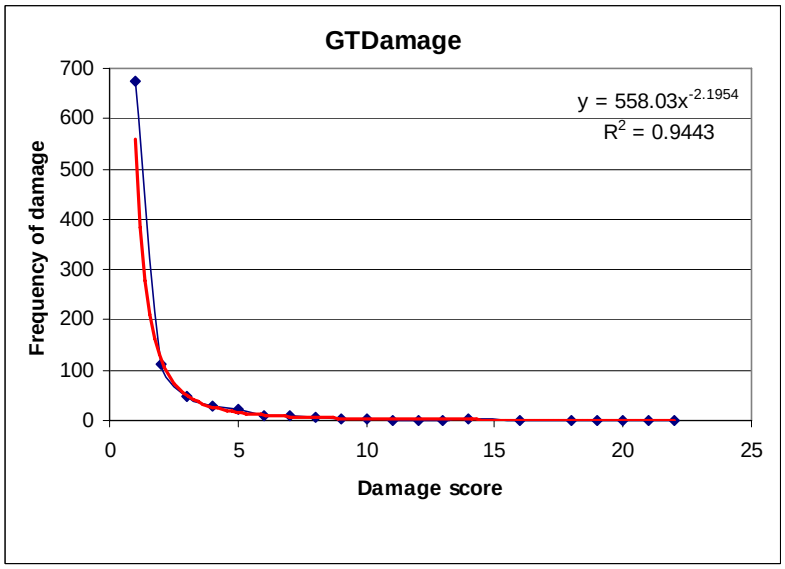


Figure 6-11 Comparison of all three damages
A. 3D scatter plot of GTDamage vs. NSDamage vs. EMDamage, each point indicates the corresponding 3 damage scores by a reaction. **B**. The 2D plot view of the GTDamage vs. NSDamage from **A**, **C**. Plot of the NSDamage vs. EMDamage, **D**. Plot of GTDamage vs. EMDamage.

To compare the three damage algorithms, GTDamage, NSDamage and EMDamage, the scores of each reaction knockout from all the small metabolic models were calculated and plotted on a 3D graph. Each coordinate represents one of the three damage score on the 3D plot. Figure 6-11A shows the 3D plot of the three damage methods, while Figure 6-11 B, C and D shows the 2D plots of two damage methods. If a reaction gives the same damage score by all three damage analysis methods (i.e. if GTDamage ≈ NSDamage ≈ EMDamage), then one would expect that all the points will be on the longer diagonal of the cube

and in case of 2D plots points will appear on the diagonal of the plot. The NSDamage and EMDamage plots show more correlation (most of the points on the diagonal) suggesting that despite the problem of non accountability of reaction direction, NSDamage prediction is closer to EMDamage. This phenomenon can be of advantage; the null space damage can be obtained for large genome scale models while there is still a problem in computation of elementary modes for EMDamage prediction for such large models

6.6.7 **Damage analysis of a large metabolic model**



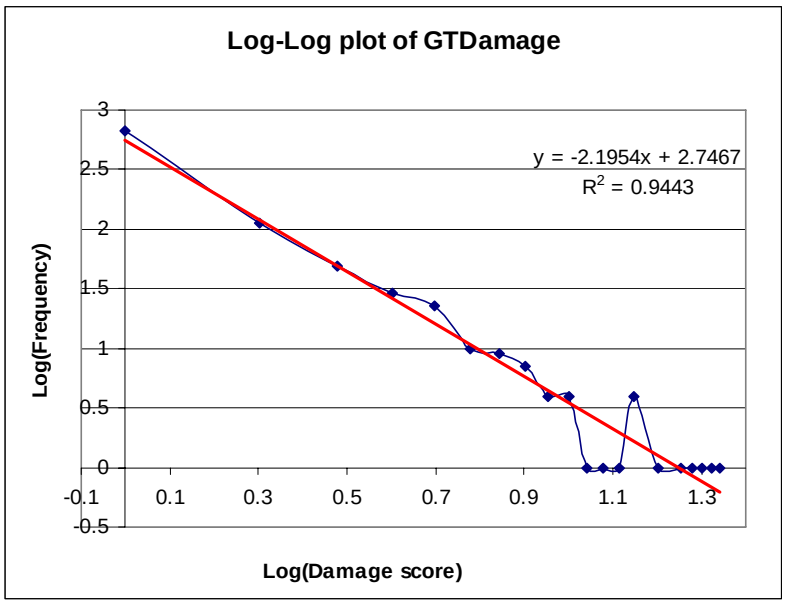


Figure 6-12 Graph theoretic damage analysis for *E.coli*-3 model

Using the modified GTDamage algorithm, analysis of the *E.coli-3* model shows that it follows the power law damage pattern as observed by (Lemke *et al.*, 2004). The power law fit shown in Figure 6-12 suggested that knockout of very few reactions have high impact of damage on the overall metabolic network. Elimination of such reactions caused extensive damage to the metabolic network, while most other reactions do not cause any significant damage.

As shown in Table 6-5, reactions with GTDamage score of 10 or more reactions were analysed for the knockout of the essential genes obtained from PEC database. Out of 17 reactions top damage score reactions, top 11 reactions knock-out at least one essential gene in *E.coli.* Five reactions which do not cause any essential gene damage directly may show some correlation with synthetic lethality (Ghim *et al.*, 2005) in *E.coli*. It will be of further interest to check the possibility any correlation of these genes with synthetic lethality data for *E.coli* in future to understand the reaction damage.

Similar results were observed by Lemke *et al.*(2004), where very few reactions showed significant damage in the metabolic network.

Table 6-5 Graph theoretic damage in E.coli-3 model (Reactions with GTDamage score of 10 or more)

	Reaction knocked	GTDamage score (>10)	Essential genes knocked (PEC database)	Total essential gene knocked
1	DXPRIi	22	b2515, b2747, b0174, b0173, b0421, b2746, b1208, b0029	8
2	MEPCT	21	b2515, b2747, b0174, b0421, b2746, b1208, b0029	7
3	CDPMEK	20	b2515, b0174, b0421, b2746, b1208, b0029	6
4	MECDPS	19	b2515, b0174, b0421, b2746, b0029	5
5	MECDPDH	18	b2515, b0174, b0421, b0029	4
6	IPDPS	16	b0174, b0421, b0029	3
7	EC-2.3.1.39-MCOATA	14	b0954, b1093, b1092, b0180, b2323	5
8	EC-6.1.1.17-GLUTRS	14	b2400	1
9	GLUTRR	14	b2400	1
10	EC-2.5.1.1-DMATT	14	b0174, b0421	2
11	EC-2.5.1.10-GRTT	13	b0174, b0421	2
12	EC-5.4.3.8-G1SATi	12	-	0
13	EC-4.2.1.24-PPBNGS	11	-	0
14	EC-4.3.1.8-HMBS	10	-	0
15	EC-1.6.4.5-TRDR	10	b2234, b2235	2
16	OCTDPS	10	-	0
17	EC-2.7.4.6-NDPK4	10	-	0

The NSDamage analysis performed on the *E.coli-3* model produced damage sets of variable size. Again as discussed in earlier section, for a same reaction, two damages differ significantly in *E.coli*. The analysis of the NSDamage shows that this algorithm simply shows damage set as enzyme subsets of the network. (i.e. reactions in an enzyme subset are obtained as damage sets in the model). This can be explained on the basis of null space;

1) null space ignores the reaction reversibility and

2) null space vectors do not show all the possible futile cycles of the system as column vectors in the null space.

Elementary mode damage could not be performed on *E.coli*-3 model because of the problems already discussed in section 2.3.5

6.7 Discussion on damage analysis

Damage analysis is one way of *in silico* identification importance of an enzyme or metabolite in a metabolic model. In context to small networks implications of damage can help in identification of reactions or metabolites important for functional enhancement of the system and may play a significant role for biochemical engineering of system or identification of drug target in the system.

In context to genomic scale metabolic network, damage analysis is a way of studying mutants of the wild type with lack of particular reaction or metabolite. Experimentally such mutants can be produced from wild type, though damage analysis provide virtual or in silico laboratory where every reaction in the virtual cell can be deleted and the properties such as viability and production capability of metabolites (primary or secondary metabolites of commercially importance) of resulting mutant can be studied.

6.8 Conclusion

The structural metabolic network can be used for *in silico* analysis of graph theoretic damage to understand the importance of individual reactions in large metabolic networks. Although such study only uses ‘local’ network properties such as neighbourhood connectivity of nodes in a network, it can still be used as

an initial choice of analysis due to its ease of computation. Improved graph theoretic algorithm functions well for some large scale metabolic networks.

Though the 'system' wide damage analysis algorithms, null space and elementary mode damage are more powerful than the graph theoretic algorithm, the former algorithm suffers due to the problem of ignoring reaction reversibility. The null space damage fails to show correct damage and returns an enzyme subset as a damage set. Elementary mode damage plays a better role in reporting the damage set. This algorithm works the best of all the three algorithms. The only problem is that this algorithm cannot be applied to genome scale metabolic networks due to the computational challenge of calculability of modes. However, this algorithm can be an important tool to understand the importance of a reaction since it uses global metabolic network properties rather than the local network connectivity of metabolites.

During the testing of NSD analysis, it was found that the algorithm suffers in two ways; reaction reversibility criteria violation and detection of internal cycles. Since one can not identify all the internal cycles present in the system just on the basis of the null space approach, it fails to give correct sets of damage reactions as expected.

The damage analysis studies can be verified by experimental methods. Systematic gene knock-out kits are available in market to identify the essential genes (and enzymes) in the cell. If an enzyme is essential, then its deletion from the system should produce a substantial damage to the metabolic network. Identification of such essential enzymes is of great importance to biotechnology and pharmaceutical industry.

Chapter 7
General discussion and future work

7.1 Metabolic network reconstruction

As structural models only need information on reaction stoichiometry and reaction reversibility, it is possible to reconstruct the 'omics' scale models of an organism's metabolism. Such models can still provide significant information using the topological properties of the metabolic network in an organism. The identification of enzyme subsets, dead-end and orphan metabolites, futile cycles, conservation relationships and elementary modes (for small structural network) can be obtained from such models. These aspects of the structural modelling can be further exploited for understanding the complex nature of the organism and used in performing *in silico* metabolic engineering studies on such organisms.

Due to the advancement of biological databases, collection and access to the large amount of biologically diverse data provides scope for further development of automated metabolic reconstruction and modelling of a complete *in silico* organism. However, as discussed in Chapter 4, there are still many challenges that need to be tackled for automated reconstruction of 'omics' scale metabolic networks. This may be due to the lack of consideration of the holistic aspect of the system-wide complexity of the biological data; the need is to establish the links from gene to enzyme to reaction to metabolite, which involves integration of the different databases that were not constructed to handle such complexity. Therefore, there is a need to tackle these problems collectively from modelling, experimental and database research communities.

To summarise, despite all the problems and challenges, structural modelling proves to be the best choice over kinetic modelling for large 'omics' scale metabolic networks.

7.2 Metabolic network substructure study

7.2.1 Subset and operons in E.coli

Reactions in a subset are 'all or none' flux carrying reactions; any change in the flux through the reaction results in proportional changes in the flux through other reactions. By analogy, this phenomenon is similar to the operonic genes in the bacterial genome. More than 15 percent of the total operons correlated to enzyme subsets in E.coli. This suggests substructure of metabolism influences genetic organisation on the genome level.

7.2.2 Metabolic network substructure and coexpression study

For 6 percent of the subsets, the genes encoding the reactions in such subsets show positive coexpression, 14 percent of the enzyme subsets show partial positive coexpression with the genes responsible for reactions in the subsets while 6 percent of the subsets do not show significant coexpression of genes responsible for reactions in the subsets.

Since a very stringent confidence test was used to identify the significance of the coexpression of genes in a subsets, and due to small size of microarray experiments (16); it may be possible to observe more subsets showing positive coexpression of genes if large amount of experimental (microarray) data are analysed with less stringent test.

The study of subsets to gene coexpression may provide further insights by identifying functionally co regulated genes in an operon or regulons. Such a study may also help in identifying regulatory genes responsible for controlling gene coexpression of genes in subsets.

Lastly, the identification of gene assignment can be varified form such a study. Since genes in a subset should be functionally coexpressed, a gene in a subset showing a weak coexpression may suggest a possibility of alternative gene or

wrong assignment of the function. Another possibility is that it may show an error or incompleteness in the metabolic model.

Some of the poor transcriptional correlations within a subset (or even strong negative correlations) reported in Chapter 5 (see Section 5.3.3) suggests further investigation is required, including experimental verification of gene to protein (enzyme) assignment.

The future plan includes the study of correlation between enzyme subsets to operon for other model organisms with published genome-scale metabolic network. (e.g. The *S. coelicolor* model published by Borodina *et al.*, (2005) developed for flux balance analysis can also be used for structural modelling.) Such models suffer from the problem of duplicate entries of a reaction (isoenzymes) and dead-end metabolites leading to the problem of dead reaction. Updating model definition and fixing the dead-end metabolites is a time consuming process as discussed in Chapter 4.

At present, reconstruction of genome scale model of *Saccharopolyspora erythraea* (Poolman and Patel, work in progress), *Streptococcus agalactiae* (Gevorgyan, work in progress) developed at CSM lab, Oxford Brookes University can also be tested for the correlation between the subsets and gene coexpression. Two pathogens, *Mycobacterium tuberculosis* and *Mycobacterium leprae* (Bonde, work in progress, University of Surrey, UK) are being studied for correlation between enzyme subsets and gene coexpression data.

The genome scale models can be used to identify similar enzyme subsets between two physiologically similar species and the knowledge gained in one organism may aid in filling gaps in other organism. It may be possible to find some correlation between commonly retained subsets across species with the cluster of orthologous genes (COGs).

Similar studies can be performed in not only eukaryotes but for various other model organisms (e.g. yeast, plants such as Arabidopsis, animal or ahuman cell metabolic models). However, the problem of compartmentations during the structural modelling in such organisms need to be tackled.

7.3 Damage analysis

The study of predicting the effects of the gene knockout in the organism has many applications such as understanding the metabolism, identification of the function of a gene, improving the yields of bioprocesses and drug (target) identification. The *in silico* gene or reaction damage may aid in prediction of such gene knockouts in model organisms. Structural metabolic models of an organism can be used to study such damage analyses.

The graph theoretic method based on metabolic network analysis may not always predict the extent of the exact damage in the organism as it considers local connectivity of the system while null space and elementary mode use system wide network properties. It was observed that damage due to elementary mode predicts the more accurate damage; however, at present the computational complexity of elementary mode limits its application on large metabolic networks. It may be possible to predict the elementary mode damage using null space damage in a large metabolic network, but improvement of the NSDamage algorithm is needed to handle reaction reversibility. However, such theoretical techniques of predicting the damage for a metabolic network need further refinement. On the other hand, the experimental verification of the results obtained from such techniques cannot be fully validated due to experimental challenges. The definition of gene essentiality is in fact very weak and changes in the external conditions and growth media etc may change the gene essentiality data.

The future work includes the study of a combination of the two damages, graph-theoretic and null space damage to obtain the correct elementary mode damage without the need for computation of elementary mode. Initially the focus of the damage analysis was on analysing reaction damage by knockout of a gene or

enzyme from the system. This can also be extended to identify a set of damaged metabolites by gene or reaction knockout.

In summary, in spite of the drastic increase in genome sequencing, it is still difficult to use the genomic data to predict and understand the organism's metabolic capacity and properties, even for a very well studied organism such as '*E.coli*'!

Bibliography

Aldous, J. M., Best, S. & Wilson, R. J. (2000). *Graphs and applications : an introductory approach.* London: Springer.

Allen, T. E., Herrgard, M. J., Liu, M. *et al.* (2003). Genome-scale analysis of the uses of the *Escherichia coli* genome: Model-driven analysis of heterogeneous data sets. *J.Bacteriol.* 185 (21), pp.6392-6399.

Allocco, D. J., Kohane, I. S. & Butte, A. J. (2004). Quantifying the relationship between co-expression, co-regulation and gene function. *BMC Bioinformatics* 5 (1), p.18.

Assmus, H. A. (2005) *Modelling the Carbohydrate Metabolism in Potato Tuber Cells* School of Biological and Molecular Sciences Oxford Brookes University Oxford

Avignone-Rossa, A., White, J., Kuiper, A. *et al.* (2002). Carbon flux distribution in antibiotic-producing chemostat cultures of *Streptomyces lividans*. *Metab.Eng.* 4 (2), p.138–150.

Ball, C. A., Awad, I. A. B., Demeter, J. *et al.* (2005). The Stanford microarray database accommodates additional microarray platforms and data formats. *Nucleic Acids Res.* 33 (Database Issue), p.D580–D582.

Barabasi, A.-L. & Albert, R. (1999). Emergence of scaling in random networks. *Science* 286, pp.509-512.

Barabasi, A.-L. & Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature Reviews-Genetics* 5 (2), pp.101-113.

Batagelj, V. & Mrvar, A. (2003). Pajek - Analysis and visualization of large networks. In: Junger, M. & Mutzel, P., eds. *Graph Drawing Software.* Springer, Berlin, pp.77-103.

Beard, D. A., Liang, S.-d. & Qian, H. (2002). Energy balance for analysis of complex metabolic networks. *Biophysical Journal* 83, p.79–86.

Becker, D., Selbach, M., Rollenhagen, C. *et al.* (2006). Robust *Salmonella* metabolism limits possibilities for new antimicrobials. *Nature* 440, pp.303-307.

Becker, S. A. & Palsson, B. O. (2005). Genome-scale reconstruction of the metabolic network in *Staphylococcus aureus* N315: an initial draft to the two-dimensional annotation. *BMC Microbiol.* 5 (1), p.8.

Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J. *et al.* (2005). GenBank. *Nucleic Acids Res.* 33 (Database issue), pp.D34-38.

Bland, M. (2001). *Multiple significance tests and the Bonferroni correction.* 3rd ed. An Introduction to Medical Statistics. Oxford University Press.

Blattner, F. R., Plunkett, G., Bloch, C. A. *et al.* (1997). The complete genome sequence of *Escherichia coli* K-12. *Science* 277 (5331), pp.1453-1474.

Bockhorst, J., Craven, M., Page, D. *et al.* (2003). A Bayesian network approach to operon prediction. *Bioinformatics* 19, p.1227–1235.

Bonde, B. K., Fell, D. A., Saudagar, P. S. *et al.* (unpublished data). *Why does threonine stimulate clavulanic acid production in Streptomyces clavuligerus?*

Bork, P. & Koonin, E. V. (1998). Predicting functions from protein sequences—where are the bottlenecks? *Nature Genetics* 18, pp.313 - 318.

Borodina, I., Krabben, P. & Nielsen, J. (2005). Genome-scale analysis of *Streptomyces coelicolor* A3(2) metabolism. *Genome Res.* 15 (6), pp.820-829.

Bouvier, J., Richaud, C., Higgins, W. *et al.* (1992). Cloning, characterization, and expression of the dapE gene of *Escherichia coli. J Bacteriol.* 174 (16), p.5265–5271.

Burgard, A. P., Nikolaev, E. V., Schilling, C. H. *et al.* (2004). Flux coupling analysis of genome-scale metabolic network reconstructions. *Genome Research* 14, pp.301-312.

Carlson, R., Fell, D. A. & Srienc, F. (2002). Metabolic Pathway Analysis of a Recombinant Yeast for Rational Strain Development. *Biotechnol. Bioeng.* 79, pp.121-134.

Carlson, R. & Srienc, F. (2004a). Fundamental *E.coli* biochemical pathways for biomass and energy production: Creation of overall flux states. *Biotechnol. Bioeng.* 86 (2), pp.149-162.

Carlson, R. & Srienc, F. (2004b). Fundamental *E.coli* biochemical pathways for biomass and energy production: Identification of reactions. *Biotechnol. Bioeng.* 85 (1), pp.1-19.

Causton, H. C., Quackenbush, J. & Brazma, A. (2003). *Microarray / Gene expression data analysis : A beginner's guide.* Blackwell Science Ltd., Oxford, UK.

Cecchini, G., Schroder, H. S. & Gunsalus, R. P. (1995). Aerobic inactivation of fumarate reductase from Escherichia coli by mutation of the [3Fe-4S]-quinone binding domain. *J Bacteriol.* 177 (16), p.4587–4592.

Chen, M. & Freier, A. (2002). Petri net based modelling and simulation of metabolic networks in the cell.

Clarke, B. L. (1981). Complete set of steady states for the general stoichiometric dynamical system. *J. Chem. Phys.* 75, pp.4970-4979.

Cornish-Bowden, A. & Hofmeyr, J.-H. S. (2002). The role of stoichiometric analysis in studies of metabolism: an example. *J. Theor. Biol.* 216, pp.179-191.

Covert, M. W., Schilling, C. H., Famili, I. *et al.* (2001). Metabolic modeling of microbial strains *in silico*. *Trends Biochem Sci.* 26 (3), pp.179-186.

Dallal, G. E. (2004). *P values*. Retrieved on 12 May 2006 from the World Wide Web: http://www.tufts.edu/~gdallal/pval.htm

Dandekar, T., Moldenhauer, F., Bulik, S. *et al.* (2003). A method for classifying metabolites in topological pathway analyses based on minimization of pathway number. *BioSystems* 70, pp.255-270.

Dandekar, T. & Sauerborn, R. (2002). Comparative genome analysis and pathway reconstruction. *Pharmacogenomics* 3. (2), pp.245-256.

Dartigalongue, C., Missiakas, D. & Raina, S. (2001). Characterization of the Escherichia coli sigma E regulon. *J Biol Chem.* 276 (24), pp.20866-20875.

Daruvar, A. d., Collado-Vides, J. & Valencia, A. (2002). Analysis of the cellular functions of *Escherichia coli* operons and their conservation in *Bacillus subtilis*. *J. Mol. Evol.* 55, p.211–221.

Draghici, S., Khatri, P., Eklund, A. C. *et al.* (2006). Reliability and reproducibility issues in DNA microarray measurements. *Trends Genetics* 22 (2), pp.101-109.

Duarte, N. C., Herrgard, M. J. & Palsson, B. O. (2004). Reconstruction and validation of *Saccharomyces cerevisiae* iND750, a fully compartmentalized genome-scale metabolic model. *Genome Research* 14, pp.298-1309.

Ebenhoh, O. & Handorf, T. (2004). *Expanding metabolic networks: a novel method for structural analysis.* Retrieved from the World Wide

Ebenhoh, O., Handorf, T. & Heinrich, R. (2004). Structural analysis of expanding metabolic networks. *Genome Informatics* 15 (1), pp.35-45.

Edwards, J. S., Ibarra, R. U. & Palsson, B. O. (2001). *In silico* predictions of *Escherichia coli* metabolic capabilities are consistent with experimental data. *Nature Biotechnology* 19, pp.125-130.

Edwards, J. S. & Palsson, B. O. (1999). Systems Properties of the *Haemophilus influenzae* Rd Metabolic Genotype. *J Biol Chem* 274 (25), pp.17410-17416.

Edwards, J. S. & Palsson, B. O. (2000). The *Escherichia coli* MG1655 *in silico* metabolic genotype: its definition, characteristics and capabilities. *PNAS* 97, pp.5528-5533.

Eisen, M. B., Spellman, P. T., Brown, P. O. *et al.* (1998). Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A* 95 (25), p.14863–14868.

Ermolaeva, M. D., White, O. & Salzberg, S. L. (2001). Prediction of operons in microbial genomes. *Nucleic Acids Research* 29 (5), pp.1216-1221.

Famili, I. & Palsson, B. O. (2003). The convex basis of the left null space of the stoichiometric matrix leads to the definition of metabolically meaningful pools. *Biophys.J.* 85 (1), pp.16-26.

Feist, A. M., Scholten, J. C. M., Palsson, B. O. *et al.* (2006). Modeling methanogenesis with a genome-scale metabolic reconstruction of *Methanosarcina barkeri*. *Molecular Systems Biology* 2, p.E1.

Fell, D. A. (1990). Substrate cycles: theoretical aspects of their role in metabolism. *Comments Theor. Biol.* 1, pp.341-357.

Fell, D. A. (1993). *The analysis of flux in substrate cycles.* In: *Modern Trends in Biothermokinetics*. New York: Plenum Press, 1993.

Fell, D. A. (1997). *Understanding the control of metabolism.* London: Portland Press.

Fell, D. A. & Small, J. R. (1986). Fat synthesis in adipose tissue. An examination of stoichiometric constraints. *Biochem. J.* 238, p.781–786.

Fell, D. A. & Wagner, A. (2000). The small world of metabolism. *Nature Biotech.* 18, pp.1121-1122.

Fleischmann, R. D., Adams, M. D., White, O. *et al.* (1995). Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science* 269 (5223), pp.496-512.

Francke, C., Siezen, R. J. & Teusink, B. (2005). Reconstructing the metabolic network of a bacterium from its genome. *Trends in Microbiology* 13 (11), pp.550-558.

Gerdes, S. Y., Scholle, M. D., Campbell, J. W. *et al.* (2003). Experimental determination and system level analysis of essential genes in *Escherichia coli* MG1655. *J Bacteriol.* 185 (19), pp.5673-5684.

Ghim, C.-M., Goh, K.-I. & Kahng, B. (2005). Lethality and synthetic lethality in the genome-wide metabolic network of *Escherichia coli*. *J. Theor. Biol.* 237, pp.401-411.

Glasner, J. D., Rusch, M., Liss, P. *et al.* (2006). ASAP: a resource for annotating, curating, comparing and disseminating genomic data. *Nucleic Acid Research* 34 (Database issue), pp.D41-D45.

Goto, S., Okuno, Y., Hattori, M. *et al.* (2002). LIGAND: database of chemical compounds and reactions in biological pathways. *Nucleic Acids Res.* 30 (1), pp.402-404.

Green, M. L. & Karp, P. D. (2005). Genome annotation errors in pathway databases due to semantic ambiguity in partial EC numbers. *Nucleic Acids Res.* 33 (13), p.4035–4039.

Hara, H., Yasuda, S., Horiuchi, K. *et al.* (1997). A promoter for the first nine genes of the Escherichia coli mra cluster of cell division and cell envelope biosynthesis genes, including ftsI and ftsW. *J Bacteriol.* 179 (18), pp.5802-5811.

Hashimoto, M., Ichimura, T., Mizoguchi, H. *et al.* (2005). Cell size and nucleoid organization of engineered *Escherichia coli* cells with a reduced genome. *Mol. Bio.* 55 (1), pp.137-149.

Hayashi, K., Morooka, N., Yamamoto, Y. *et al.* (2006). Highly accurate genome sequences of *Escherichia coli K-12* strains MG1655 and W3110. *Molecular Systems Biology* 2.

Hefferon, J. (2003). *Linear algebra.* Mathematics, Saint Michael's College, VT USA, 05439.

Heinemann, M., Kummel, A., Ruinatscha, R. *et al.* (2005). *In silico* genome-scale reconstruction and validation of the *Staphylococcus aureus* metabolic network. *Biotechnol Bioeng.* 92 (7), pp.850-864.

Heinrich, R., Rapoport, S. M. & Rapoport, T. A. (1977). Metabolic regulation and mathematical models. *Prog. Biophys. Molec. Biol.* 32, pp.1-82.

Heinrich, R. & Schuster, S. (1996). *The regulation of cellular systems.* London, England: Chapman & Hall.

Hofmeyr, J.-H. & Cornish-Bowden, A. (1997). The reversible Hill equation: how to incorporate cooperative enzymes into metabolic models. *Comput Appl Biosci.* 13 (4), p.377–385.

Hofmeyr, J.-H. S. (1986). Steady state modelling of metabolic pathways: a guide to the prospective simulator. *Comp. Appl. Biosci.* 2, pp.5-11.

Holme, P., Huss, M. & Jeong, H. (2003). Subnetwork hierarchies of biochemical pathways. *Bioinformatics* 19, pp.532-538.

Hommel, G. (1988). A stagewise rejective multiple test procedure based on a modified Bonferroni test. *Biometrika* 75 (2), pp.383-386.

Hong, S. H., Kim, J. S., Lee, S. Y. *et al.* (2004). The genome sequence of the capnophilic rumen bacterium *Mannheimia succiniciproducens*. *Nat Biotechnol.* 22 (10), pp.1275-1281.

Ibarra, R. U., Edwards, J. S. & Palsson, B. O. (2002). *Escherichia coli* K-12 undergoes adaptive evolution to achieve *in silico* predicted optimal growth. *Nature* 420, pp.186-189.

Ihmels, J., Levy, R. & Barkai, N. (2004). Principles of transcriptional control in the metabolic network of *Saccharomyces cerevisiae*. *Nature Biotech.* 22 (1), pp.86-92.

Imielinski, M., Belta, C., Halasz, A. *et al.* (2005). Investigating metabolite essentiality through genome-scale analysis of *Escherichia coli* production capabilities. *Bioinformatics* 21 (9), pp.2008-2016.

Imielinski, M., Belta, C., Rubin, H. *et al.* (2006). Systematic analysis of conservation relations in *E. coli* genome-scale metabolic network reveals novel growth media. *Biophys. J.* 90, pp.2659-2672.

Itoh, T., Takemoto, K., Mori, H. *et al.* (1999). Evolutionary instability of operon structures disclosed by sequence comparisons of complete microbial genomes. *Mol Biol Evol* 16, pp.332-346.

Jacob, F. & Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *J Mol Biol.* 3, pp.318--356.

Jeong, H., Tombort, B., Albert, R. *et al.* (2000). The large-scale organization of metabolic networks. *Nature* 407, pp.651 - 654.

Joshi-Tope, G., Gillespie, M., Vastrik, I. *et al.* (2005). Reactome: a knowledgebase of biological pathways. *Nucleic Acids Res.* 33 (Database Issue), pp.D428-D432.

Kanehisa, M., Goto, S., Hattori, M. *et al.* (2006). From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res. 34, D354-357 (2006).* 34, pp.D354-357.

Karp, P. D. (2006). *E. coli K-12 Transcription Unit: ilvLG_1G_2MEDA*. Retrieved on May 2005 from the World Wide Web: http://biocyc.org/ECOLI/NEW-IMAGE?type=OPERON&object=TU524

Karp, P. D., Ouzounis, C. A., Moore-Kochlacs, C. *et al.* (2005). Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acid Research* 33 (19), pp.6083-6089.

Karp, P. D., Paley, S. & Romero, P. (2002a). The pathway tools software. *Bioinformatics* 18, pp.S225-232.

Karp, P. D., Riley, M., Saier, M. *et al.* (2002b). The EcoCyc database. *Nucleic Acids Res.* 30 (1), pp.56-58.

Karp, P. D., Riley, M., Saier, M. *et al.* (2000). The EcoCyc and MetaCyc databases. *Nucleic Acids Res.* 28 (1), pp.56-59.

Kaw, A. K. (2002). *Introduction to matrix algebra*. Retrieved on May 12 2005 from the World Wide Web: http://numericalmethods.eng.usf.edu/matrixalgebrabook/frmMatrixDL.asp

Kharchenko, P., Church, G. M. & Vitkup, D. (2005). Expression dynamics of a cellular metabolic network. *Molecular Systems Biology*.

Kholodenko, B. N., Shuster, S., Rohwer, J. M. *et al.* (1995). Composite control of cell function: Metabolic pathways behaving as single control units. *FEBS Lett.* 368, pp.1-4.

Kim, J. S. & Lee, S. Y. (2001). *In silico* metabolic pathway modeling and analysis of *Mycoplasma pneumoniae*. *Genome Informatics* 12, pp.298-299.

Klamt, S. (2006). Generalized concept of minimal cut sets in biochemical networks. *BioSystems* 83 (2-3), pp.233-247.

Klamt, S. & Gilles, E. D. (2004). Minimal cut sets in biochemical reaction networks. *Bioinformatics* 20 (2), pp.226-234.

Klamt, S., Schuster, S. & Gilles, E. D. (2002a). Calculability analysis in underdetermined metabolic networks illustrated by a model of the central

metabolism of in purple nonsulphur metabolism. *Biotechnol. Bioeng.* 77 (7), pp.734-750.

Klamt, S., Stelling, J. & Ginkel, M. (2002b). *Pathway analysis in metabolic networks: combinatorial complexity and an efficient software platform*. In: *Proceedings of the 3rd International Conference on Systems Biology*, 2002b.

Klamt, S., Stelling, J., Ginkel, M. *et al.* (2003). FluxAnalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. *Bioinformatics* 19 (2), pp.261-269.

Lederberg, J. & Tatum, E. L. (1946). Gene recombination in *Escherichia coli*. *Nature* 158, p.558.

Lee, H. K., Hsu, A. K., Sajdak, J. *et al.* (2004). Coexpression analysis of human genes across many microarray data sets. *Genome Research* 14 (6), pp.1085-1094.

Lehninger, A. L., Nelson, D. L. & Cox, N. N. (1993). *Principles of biochemistry*. New York: Wiley, 1993.

Leiser, J. & Blum, J. J. (1987). On the analysis of substrate cycles in large metabolic systems. *Cell Biophys.* 11, pp.123-138.

Lemke, N., Heredia, F., Barcellos, C. a. K. *et al.* (2004). Essentiality and damage in metabolic networks. *Bioinformatics* 20 (1), p.115 119.

Lesk, A. M. (2002). *Introduction to bioinformatics*. Oxford University Press.

Lespinet, O. & Labedan, B. (2005). Orphan enzymes? *Science* 307, p.42.

Li, W. (2006). *Bibliography on Microarray Data Analysis*. Retrieved on Feb 17th 2006 from the World Wide Web: http://www.nslij-genetics.org/microarray/

Lutz, M. (2001). *Programming Python.* O'Reilly & Associates.

Lutz, M. & Ascher, D. (1999). *Learning Python.* O'Reilly & Associates.

Ma, H. & Zeng, A.-P. (2003). Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics* 19 (2), pp.270-277.

Maas, W. K. (1964). Studies on the mechanism of repression of arginine biosynthesis in *Escherichia coli*. II. Dominance of repressibility in diploids. *J Mol Biol.* 78, pp.365-370.

Maltsev, N., Glass, E., Sulakhe, D. *et al.* (2006). PUMA2—grid-based high-throughput analysis of genomes and metabolic pathways. *Nucleic Acids Res* 34 (Database Issue), pp.D369-D372.

Meloni, R., Khalfallah, O. & Biguet, N. F. (2004). DNA microarrays and pharmacogenomics. *Pharmacological Research* 49, pp.303\-308.

Mendes, P. (1993). Gepasi - a software package for modeling the dynamics, steady-states and control of biochemical and other systems. *Comp. Appl. Biosci.* 9 (5), pp.563-571.

Michal, G. (1998). On representation of metabolic pathways. *BioSystems* 47, pp.1-7.

Michal, G., ed. (1999). *Biochemical pathways : an atlas of biochemistry and molecular biology*. New York ; Chichester : Wiley, c1999.

Monod, J. (1972). *L'éléphant et l'Escheria Coli*. Retrieved on Dossier 24.2 20 from the World Wide Web: http://www.pasteur.fr/infosci/archives/mon/im_ele.html

Nadon, R. & Shoemaker, J. (2002). Statistical issues with microarrays: processing and analysis. *Trends Genetics* 18 (5), pp.265-271.

Naumoff, D. G., Xu, Y., Glansdorff, N. *et al.* (2004). Retrieving sequences of enzymes experimentally characterized but erroneously annotated : the case of the putrescine carbamoyltransferase. *BMC Genomics* 5, p.52.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review* 45, pp.167-256.

Nielsen, J. (1998). Metabolic engineering: techniques for analysis of targets for genetic manipulations. *Biotech. Bioeng.* 58, pp.125-132.

Ogura, M., Yamaguchi, H., Yoshida, K.-i. *et al.* (2001). DNA microarray analysis of *Bacillus subtilis* DegU, ComA and PhoP regulons: an approach to comprehensive analysis of *B. subtilis* two-component regulatory systems. *Nucleic Acids Research* 29 (18), pp.3804-3813.

Okuda, S., Katayama, T., Kawashima, S. *et al.* (2006). ODB: a database of operons accumulating known operons across multiple genomes. *Nucleic Acids Research* 34 (Database Issue), pp.D358-D362.

Oliveira, A. P., Nielsen, J. & Forster, J. (2005). Modeling *Lactococcus lactis* using a genome-scale flux model. *BMC Microbiol.* 5, p.39.

Olivier, B. G., Rohwer, J. M. & Hofmeyr, J.-H. S. (2005). Modelling cellular systems with PySCeS. *Bioinformatics* 21 (4), pp.560-561.

Osterman, A. & Overbeek, R. (2003). Missing genes in metabolic pathways: a comparative genomics approach. *Current Opinion in Chemical Biology* 7, pp.238\-251.

Pairwise comparisons in SAS and SPSS: Bonferroni-Holm and Sidak-Holm test. (2005). Retrieved on 20 Dec 2005 from the World Wide Web: http://www.uky.edu/ComputingCenter/SSTARS/MultipleComparisons_3.htm

Paley, S. M. & Karp, P. D. (2002). Evaluation of computational metabolic-pathway predictions for *Helicobacter pylori*. *Bioinformatics* 18 (5), pp.715-724.

Papin, J. A., Price, N. D. & Palsson, B. O. (2002). Extreme pathway lengths and reaction participation in genome-scale metabolic networks. *Genome Research* 12 (12), pp.1889-1900.

Papin, J. A., Price, N. D., Wiback, S. J. *et al.* (2003). Metabolic pathways in the post-genome era. *Trends Biochem. Sci.* 28 (5), pp.250-258.

Patil, K. R. & Nielsen, J. (2005). Uncovering transcriptional regulation of metabolism by using metabolic network topology. *Proc Natl Acad Sci* 102 (8), pp.2685-2689.

Pfeiffer, T., Sanchez-Valdenebro, I., Nuno, J. C. *et al.* (1999). Metatool: for studying metabolic networks. *Bioinformatics* 15 (3), pp.251-257.

Poolman, M. (2004). *PyoCyc: ScrumPy module for building genome-scale metabolic model using MetaCyc database (Unpublished data)*.

Poolman, M. G. (2006a). ScrumPy - Metabolic Modelling in Python. *IEE Proc. Systems Biology* 153 (5), pp.375-378.

Poolman, M. G. (2006b). *ScrumPy - Metabolic Modelling in Python, software and manual*. Retrieved from the World Wide Web: http://mudshark.brookes.ac.uk/ScrumPy

Poolman, M. G., Assmus, H. E. & Fell, D. A. (2004a). Applications of metabolic modelling to plant metabolism. *J.Exp.Bot.* 55 (400), pp.1177-1186.

Poolman, M. G., Bonde, B. K., Gevorgyan, A. *et al.* (2006). Challenges to be faced in the reconstruction of metabolic networks from public databases. *IEE Proc. Systems Biology* 153 (5), pp.379-384.

Poolman, M. G., Fell, D. A. & Raines, C. A. (2003). Elementary modes analysis of photosynthate metabolism in the chloroplast stroma. *Eur. J. Biochem* 270, pp.430-439.

Poolman, M. G., Venkatesh, K. V., Pidcock, M. K. *et al.* (2004b). A method for the determination of flux in elementary modes, and its application to *Lactobacillus rhamnosus*. *Biotechnol. Bioeng.* 88 (5), pp.601-612.

Pramanik, J. & Keasling, J. D. (1997). Stoichiometric model of *Escherichia coli* metabolism: incorporation of growth-rate dependent biomass composition and mechanistic energy requirements. *Biotechnol. Bioeng.* 56, pp.398-421.

Press, W., Flannery, B., Teukolsky, S. *et al.* (1989). Numerical Recipes in C. Cambridge: Cambridge University Press.

Reder, C. (1988). Metabolic control theory: a structural approach. *J. Theor. Biol.* 135, p.175–201.

Reed, J. L. & Palsson, B. O. (2004). Genome-scale *in silico* models of *E.coli* have multiple equivalent phenotypic states: assessment of correlated reaction subsets that comprise network states. *Genome Research* 14, pp.1797-1805.

Reed, J. L., Vo, T. D., Schilling, C. H. *et al.* (2003). An expanded genome-scale model of *E.coli* K-12 (iJR904 GSM/GPR). *Genome Biology* 4, p.R54.

Riley, M. (1993). Functions of the gene products of *Escherichia coli*. *Microbiol Rev.* 57 (4), pp.862-952.

Riley, M., Abe, T., Arnaud, M. B. *et al.* (2006). *Escherichia coli* K-12: a cooperatively developed annotation snapshot - 2005. *Nucleic Acid Research* 34 (1), pp.1-9.

Rohwer, J. M., Schuster, S. & Westerhoff, H. V. (1996). How to recognize monofunctional units in a metabolic system. *J.Theor.Biol.* 179 (3), pp.213-228.

Ross-Macdonald, P., Coelho, P. S., Roemer, T. *et al.* (1999). Large-scale analysis of the yeast genome by transposon tagging and gene disruption. *Nature* 402, p.413–418.

Salgado, H., Gama-Castro, S., Martínez-Antonio, A. *et al.* (2004). RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in Escherichia coli K-12. *Nucleic Acids Res.* 32 (Database issue), pp.D303-306.

Salgado, H., Moreno-Hagelsieb, G., Smith, T. F. *et al.* (2000). Operons in Escherichia coli: genomic analyses and predictions. *Proc. Natl Acad. Sci.* 97, p.6652–6657.

Sauro, H. M. (1986) *Control analysis and simulation of metabolism* Sch. Bio. Mol. Sciences Oxford Polytechnic Oxford

Sauro, H. M. (1993). SCAMP: a general-purpose simulator and metabolic control analysis program. *Comput. Applic. Biosci.* 9, pp.441-450.

Sauro, H. M. (2000). *Jarnac: a system for interactive metabolic analysis*. In: *Animating the cellular map: Proceedings of the 9th Int. meeting on BioThermoKinetics*: Stellenbosch University Press,, 2000.

Sauro, H. M. & Fell, D. A. (1991). SCAMP: a metabolic simulator and control analysis program. *Mathl. Comput. Modelling* 15, pp.15-28.

Sauro, H. M. & Ingalls, B. P. (2004). Conservation analysis in biochemical networks: computational issues for software writers. *Biophysical Chemistry* 109, pp.1-15.

Schena, M., Shalon, D., Davis, R. W. *et al.* (1995). Quantitative monitoring of gene-expression patterns with a complementary-DNA microarray. *Science* 270 (5235), pp.467-470.

Schilling, C. H., Covert, M. W., Famili, I. *et al.* (2002). Genome-scale metabolic model of *Helicobacter pylori* 26695. *J. Bacteriology* 184 (16), pp.4582-4593.

Schilling, C. H., Letscher, D. & 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, pp.229-248.

Schilling, C. H., Schuster, S., Palsson, B. O. *et al.* (1999). Metabolic pathway analysis: basic concepts and scientific applications in the post-genomic era. *Biotechnol. Prog.* 15, pp.296-303.

Schomburg, I., Chang, A., Ebeling, C. *et al.* (2004). BRENDA, the enzyme database: updates and major new developments. *Nucleic Acids Res.* 32 (Database issue), pp.D431-433.

Schreiber, F. (2003). *Comparison of metabolic pathways using constraint graph drawing*. In: *Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics*. Adelaide, Australia: Australian Computer Society, Inc., 2003.

Schuler, G. D. (2001). Sequence alignment and database searching. In: Baxevanis, A. D. & Ouellette, B. F. F., eds. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. 2nd ed. J. Wiley & Sons, pp.187-212.

Schuster, R. & Schuster, S. (1993). Refined algorithm and computer program for calculating all non-negative fluxes admissible in steady states of biochemical reaction systems with or without some flux rates fixed. *Comput. Appl. Biosci.* 9 (1), pp.79-85.

Schuster, S., Dandekar, T. & Fell, D. A. (1999). Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends. Biotech.* 17 (2), pp.53-60.

Schuster, S., Fell, D. A. & Dandekar, T. (2000). A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotech.* 18, pp.326-332.

Schuster, S. & Hilgetag, C. (1994). On elementary flux modes in biochemical systems at steady state. *J. Biol. Syst.* 2, pp.165-182.

Schuster, S. & Hilgetag, C. (1995). What information about the conserved-moiety structure of chemical reaction systems can be derived from their stoichiometry? *J. Phys. Chem.* 99, pp.8017-8023.

Schuster, S., Hilgetag, C. & Fell, D. A., eds. (1994). *Detecting elementary modes of functioning in metabolic networks*. Modern Trends in Biothermokinetics. Innsbruck University Press, Innsbruck.

Schuster, S., Hilgetag, C., Woods, J. H. *et al.* (2002a). Reaction routes in biochemical reaction systems: algebraic properties, validated calculation procedure and example from nucleotide metabolism. *J. Math. Biol.* 45, pp.153-181.

Schuster, S., Klamt, S., Weckwerth, W. *et al.* (2002b). Use of network analysis of metabolic systems in bioengineering. *Bioprocess Biosys. Eng.* 24, pp.363-373.

Schuster, S., Pfeiffer, T., Moldenhauer, F. *et al.* (2002c). Exploring the pathway structure of metabolism: decomposition into subnetworks and application to *M. pneumoniae*. *Bioinformatics* 18 (2), pp.351-361.

Schuster, S. & Schuster, R. (1991). Detecting strictly detailed balanced subnetworks in open chemical-reaction networks. *J.Math.Chem.* 6 (1), pp.17-40.

Schwarz, R., Musch, P., Kamp, A. v. *et al.* (2005). YANA – a software tool for analyzing flux modes, gene-expression and enzyme activities. *BMC Bioinformatics* 6, p.135.

Segre, D., Vitkup, D. & Church, G. M. (2002). Analysis of optimality in natural and perturbed metabolic networks. *Proc Natl Acad Sci U S A* 99 (23), pp.15112-15117.

Serres, M. H., Gopal, S., Nahum, L. A. *et al.* (2001). A functional update of the *Escherichia coli* K-12 genome. *Genome Biology* 2 (9), pp.1-7.

Sheikh, K., Forster, J. & Nielsen, L. K. (2005). Modeling hybridoma cell metabolism using a generic genome-scale metabolic model of *Mus musculus*. *Biotechnol Prog.* 21 (1), pp.112-121.

Sirko, A., Hryniewicz, M., Hulanicka, D. *et al.* (1990). Sulfate and thiosulfate transport in Escherichia coli K-12: nucleotide sequence and expression of the cysTWAM gene cluster. *J Bacteriol.* 1990 (6), pp.3351-3357.

Sokal, R. R. & Rohlf, F. J. (1995). *Biometry: principles and practice of statistics in biological research.* W.H.Freeman & Co Ltd.

Steinhauser, D., Junker, B. H., Luedemann, A. *et al.* (2004a). Hypothesis-driven approach to predict transcriptional units from gene expression data. *Bioinformatics* 20 (12), pp.1928-1939.

Steinhauser, D., Usadel, B., Luedemann, A. *et al.* (2004b). CSB-DB A comprehensive systems-biology database. *Bioinformatics* 20 (18), pp.3647-3651.

Stelling, J., Klamt, S., Bettenbrock, B. *et al.* (2002). Metabolic network structure determines key aspects of functionality and regulation. *Nature* 420, pp.190-193.

Stephanopoulos, G. N., Aristidou, A. A. & Nielsen, J. (1998). *Metabolic engineering principles and methodologies.* London, U.K.: Academic Press.

Stryer, L. (1995). *Biochemistry.* New York, U.S.A.: W. H. Freeman.

Stuart, J. M., Segal, E., Koller, D. *et al.* (2003). A gene-coexpression network for global discovery of conserved genetic modules. *Science* 302 (5643), pp.249-255.

Tatusov, R. L., Fedorova, N. D., Jackson, J. D. *et al.* (2003). The COG database: an updated version includes eukaryotes. *BMC Bioinformatics* 4 (1), pp.41-54.

Tatusov, R. L., Galperin, M. Y., Natale, D. A. *et al.* (2000). The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res.* 28 (1), pp.33-36.

Teusink, D. B. (2005). *A genome-scale model of Lactobacillus plantarum: useful for omics data integration and exploring metabolic capacities, but less so for flux predictions.* Retrieved on 10 Dec 2005 from the World Wide Web: http://www.iu-bremen.de/news/events/07077/

Thomason, L. C., Court, D. L., Datta, A. R. *et al.* (2004). Identification of the Escherichia coli K-12 ybhE gene as pgl, encoding 6-phosphogluconolactonase. *J. Bacteriol.* 186 (24), pp.8248-8253.

TIGR MeV MultiExperiment viewer version 3.1. (2005). Retrieved on June 20 2005 from the World Wide Web: http://www.tm4.org/mev.html

Tjaden, B., Haynor, D. R., Stolyar, S. *et al.* (2002a). Identifying operons and untranslated regions of transcripts using *Escherichia coli* RNA expression analysis. *Bioinformatics* 18 (1), pp.s337-s344.

Tjaden, B., Saxena, R. M., Stolyar, S. *et al.* (2002b). Transcriptome analysis of *Escherichia coli* using high-density oligonucleotide probe arrays. *Nucleic Acids Res* 30 (17), pp.3732-3738.

Tsoka, S., Simon, D. & Ouzounis, C. A. (2004). Automated metabolic reconstruction for *Methanococcus jannaschii*. *Archaea* 1, p.223–229.

Urbanczik, R. & Wagner, C. (2005). An improved algorithm for stoichiometric network analysis: theory and applications. *Bioinformatics* 21 (7), pp.1203-1210.

Van der Heijden, R. T. J. M., Romein, B., Heijnen, J. J. *et al.* (1994). Linear constraint relations in biochemical reaction system: I. Classification of the calculability and the balanceability of conversion rates. *Biotech Bioeng* 43, pp.3-10.

Varma, A., Boesch, B. W. & Palsson, B. O. (1993). Biochemical production capabilities of *Escherichia coli*. *Biotech. Bioeng.* 42, pp.59-73.

Varma, A. & Palsson, B. O. (1994). Metabolic flux balancing: basic concepts, scientific and practical use. *Bio/Technology* 12, pp.994-998.

Wagner, A. & Fell, D. A. (2001). The small world inside large metabolic networks. *Proc. Roy. Soc. London B* 268, pp.1803-1810.

Wagner, C. (2004). Nullspace approach to determine elementary modes of chemical reaction systems. *J. Phys. Chem. B* 108 (7), pp.2425-2431.

Warrington, J. A., Todd, R. & Wong, D., eds. (2002). *Microarrays and cancer research.* Westboro, MA: BioTechniques Press.

Watts, D. J. & Strogatz, S. H. (1998). Collective dynamics of 'small world' network. *Nature* 393, pp.440-442.

Werner-Washburne, M., Wylie, B., Boyack, K. *et al.* (2002). Comparative Analysis of Multiple Genome-Scale Data Sets. *Genome Res.* 12 (10), pp.1564-1573.

Wiechert, W. (2001). 13C Metabolic flux analysis. *Metabolic Engineering* 3 (3), pp.195-206.

Wilhelm, T., Behre, J. & Schuster, S. (2004). Analysis of structural robustness of metabolic networks. *Systems Biology* 1, pp.114 - 120.

Winzeler, E. A., Shoemaker, D. D., Astromoff, A. *et al.* (1999). Functional characterization of the *S. cerevisiae* genome by gene deletion and parallel analysis. *Science* 285 (5429), p.901–906.

Wolfe, C. J., Kohane, I. S. & Butte, A. J. (2005). Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks. *BMC Bioinformatics* 6, p.227.

Yeh, I., Hanekamp, T., Tsoka, S. *et al.* (2004). Computational analysis of *Plasmodium falciparum* metabolism: organizing genomic information to facilitate drug discovery. *Genome Research* 14 (5), pp.917-924.

Zevedei-Oancea, I. & Schuster, S. (2003). Topological analysis of metabolic networks based on Petri net theory. *In Silico Biology* 3, p.29.

Appendix A

Matrix algebra and structural metabolic modelling

Matrix algebra is used to solve a system of simultaneous linear equations (Kaw, 2002). A system of m linear equations and n unknown is of the following form,

$$a_{11}x_1 + a_{12}x_2 + \cdots\cdots + a_{1n}x_n = C_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots\cdots + a_{2n}x_n = C_2$$
$$a_{31}x_1 + a_{32}x_2 + \cdots\cdots + a_{3n}x_n = C_3$$
$$\cdots\cdots$$
$$\cdots\cdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots\cdots + a_{mn}x_n = C_m$$

Above system can also be written in the matrix form as,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \cdots & \cdots & a_{2n} \\ a_{31} & a_{31} & \cdots & \cdots & \cdots & a_{31} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m1} & \cdots & \cdots & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ \vdots \\ C_m \end{bmatrix}$$

Denoting the above three matrices as A, X and C, the system of equation is AX=C, where A is called coefficient matrix, vector X is called solution (unknown) vector and C is called right hand side vector.

If the above concept is applied to metabolic system at steady state, the stoichiometry matrix represents a coefficient matrix, $\bar{v}$ is a vector of rate of all reactions in the system (represents X) and vector C, the right hand side vector will represent a zero vector as the system is at steady state,

$$\boldsymbol{N}\bar{v} = \boldsymbol{0} \qquad (14)$$

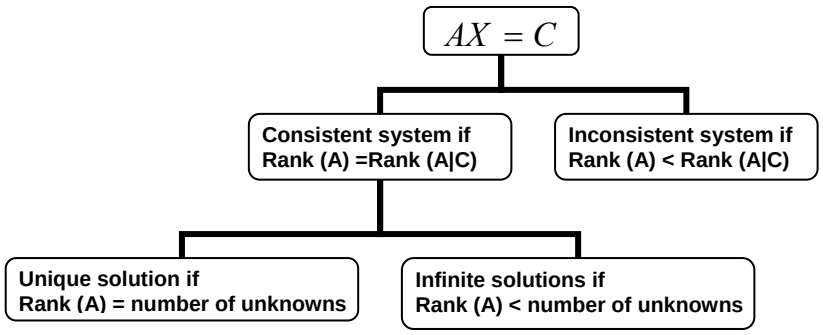
Rank of a Matrix

The rank of a matrix is the dimension of the matrix, corresponding to the number of linearly independent rows or columns of the matrix. For a matrix [A], a rank is determined as the order of the largest square sub matrix whose determinant is not zero and is denoted as rank(A).

Solution of system

In the system of equations AX = C, the system is consistent if there is a solution and inconsistent (over determined system) if no solution is possible. Consistent system may have only one unique solution or infinite solutions (underdetermined system). Rank of the matrix can be used to determine the consistent system. If the rank of the coefficient matrix A is same as the rank of augmented matrix [A|C], then it is a consistent system of equations and if rank(A) is less than rank(A|C) then it is an inconsistent system.

If in addition, the rank of the coefficient matrix [A] is same as the number of unknowns, then the solution is unique; if the rank of the coefficient matrix [A] is less than the number of unknowns, then infinite solutions exist.



Appendix Figure 1 The possible solutions of the system

Degree of freedom (F)

The degree of freedom the stoichiometric matrix determines how many fluxes that must be known to fully resolve the metabolic fluxes in the network. The number of degrees of freedom gives the dimension of the null space.

If ν_n is the total number of fluxes (i.e. the dimension of the vector ν) then the degree of freedom (F) is

$$F = \nu_n - rank(N) \qquad (15)$$

where rank(N) is the rank of N, F is the number of degree of freedom in the system.

Row echelon form of a matrix:

For the row echelon form, an identity matrix of m (number of rows) dimension is placed side by side to the stoichiometry matrix to obtain a new augmented matrix. Each column of identity matrix refers to a different time derivative.
Gaussian elimination is performed to get the row echelon form matrix. It should as far as possible, has non-zero elements on the main diagonal, with only zero elements below. In practice, some times it is impossible to have such an ideal non-zero diagonal element matrix, so the diagonal is allowed to shift to right. The elements on the corner of such shifted diagonal are called pivots and they must be nonzero.

Reduced stoichiometry matrix (row echelon form)

If all the rows (m) of N are linearly independent, then rank (N) = m. If rank (N) < m then there are m-r dependencies among the rows or in the differential equations of system. Eliminating m-r dependent rows of N gives a new form of a matrix called as reduced stoichiometry matrix N_R.
Reduced stoichiometry matrix must be in row echelon form and has rank (N) independent rows. N and N_R can be related with a link matrix (L),

$$\mathrm{N} = \mathrm{L}\mathrm{N}_\mathrm{R} \tag{16}$$

So Eq. 2.4 becomes,

$$\mathrm{L}\mathrm{N}_\mathrm{R}\, \nu = 0 \tag{17}$$

If N is rearranged so that the independent rows are on top and dependent rows are at the bottom of the N, the corresponding rearrangement in L and the concentration vector ν can be written as,

$$L = \begin{bmatrix} I \\ L_o \end{bmatrix} \text{ and } \nu = \begin{bmatrix} \nu_i \\ \nu_d \end{bmatrix} \tag{18}$$

where I is an identity matrix, ν_I is independent flux vector and ν_d is dependent flux vector. L_o matrix is used to analyse the conserved moieties in the network.

Appendix B

Definitions of genomic terms

Gene- A gene is the segment of DNA involved in producing a polypeptide chain or stable RNA.

Structural Gene - is defined a gene that controls the production of a specific protein or peptide.

Regulatory Gene – This is a gene which controls the protein-synthesizing activity of other genes in an operon.

Operon – Operon is a group of genes that regulate the function of other genes by affecting the synthesis of mRNA

Repressor – Repressor is a protein produced by a regulatory gene which binds to a site on the operon and prevents transcription of structural genes

Inducer – Inducer is a metabolite which prevents the repressor from binding to the DNA so that the structural genes can be transcribed.

Operator – Operator is a site at which the repressor binds.

Transcription unit - A transcription unit (TU) is a set of one or more genes transcribed from a single promoter. A TU may also include regulatory protein binding sites affecting this promoter and a terminator.

Homolog- Homolog is a gene which can be related to the second gene originated from a common ancestral DNA sequence.

Ortholog- Orthologs are genes in different species that evolved from a common ancestral gene by speciation. Normally, orthologs retain the same function in the course of evolution. Identification of orthologs is critical for reliable prediction of gene function in newly sequenced genomes.

Paralog - Paralogs are genes related by duplication within a genome. Orthologs retain the same function in the course of evolution, whereas paralogs evolve new functions, even if these are related to the original one.

Appendix C

Example of the Bonferroni-Holm test for gene pairs in a subset

This example subset gives test procedure for Bonferroni test performed on the pairs of genes in a subset.

Subset 223 (see Section 5.3.3 B1 for more details) which associates 10 reactions to 11 genes results in 55 pair wise expression (Pearson's correlation coefficient) collected from 16 array plates.

The Bonferroni-Holm test procedure (as explained in Section 5.1.2) was applied to these 55 pairs to identify significant pairs. As explained in the text, Pearson's correlation coefficient (R) and original p-value was obtained from CSB.DB Co-response matrix.

All the 55 pairs were arranged in increasing order of p-values and ranked in a reverse order. The modified p-value (p-mod) is obtained for each pair by dividing the alpha (i.e. 0.05) by rank.

Original p-value is then compared with modified p-value to decide confidence of the association:

H_0 (gene pair associates by chance): accept null hypothesis if original p-value is greater than modified p-value, else accept alternative hypothesis (Gene pair in a subset are significant).

If a pair with null hypothesis acceptable is found, stop further testing and accept null hypothesis for all the remaining pairs.

Pair number	Gene1	Gene2	Pearson's Correlation coefficient (R)	Original p-value (p)	Modified P-value (p-mod) (=alpha/rank)	rank	Bonferroni test:- Accept H_0 if True i.e. (p> p-mod)
1	b0754	b0908	0.969021	6.77E-10	0.000909	55	FALSE
2	b3390	b0388	0.937966	8.04E-08	0.000926	54	FALSE
3	b1704	b0754	0.932947	1.37E-07	0.000943	53	FALSE
4	b1704	b0908	0.921047	4.15E-07	0.000962	52	FALSE
5	b2329	b3390	0.901061	1.91E-06	0.00098	51	FALSE
6	b1693	b3390	0.898203	2.31E-06	0.001	50	FALSE
7	b0754	b2329	0.892901	3.25E-06	0.00102	49	FALSE
8	b0754	b1693	0.886534	4.79E-06	0.001042	48	FALSE
9	b2601	b3389	0.885757	5.01E-06	0.001064	47	FALSE
10	b1693	b0908	0.884723	5.32E-06	0.001087	46	FALSE
11	b2329	b0388	0.879957	6.97E-06	0.001111	45	FALSE
12	b2601	b0388	0.859786	1.96E-05	0.001136	44	FALSE
13	b0388	b0908	0.851450	2.86E-05	0.001163	43	FALSE
14	b2329	b0908	0.849300	3.15E-05	0.00119	42	FALSE
15	b0754	b3390	0.846524	3.55E-05	0.00122	41	FALSE
16	b3390	b0908	0.845610	3.69E-05	0.00125	40	FALSE
17	b1693	b2329	0.838449	4.97E-05	0.001282	39	FALSE

18	b1693	b3389	0.818138	0.000108	0.001316	38	FALSE
19	b3389	b3390	0.815463	0.000118	0.001351	37	FALSE
20	b0754	b0388	0.814827	0.000121	0.001389	36	FALSE
21	b3389	b0388	0.805310	0.000167	0.001429	35	FALSE
22	b1704	b2329	0.798884	0.000206	0.001471	34	FALSE
23	b1693	b0388	0.797439	0.000216	0.001515	33	FALSE
24	b2601	b3390	0.766445	0.000534	0.001563	32	FALSE
25	b1704	b0388	0.765430	0.000549	0.001613	31	FALSE
26	b1704	b1693	0.742979	0.000975	0.001667	30	FALSE
27	b1704	b3390	0.742053	0.000997	0.001724	29	FALSE
28	b3389	b0908	0.725972	0.001452	0.001786	28	FALSE
29	b2601	b1693	0.704896	0.002292	0.001852	27	TRUE
30	b2601	b0908	0.692763	0.002931	0.001923	26	TRUE
31	b3389	b2329	0.684065	0.003473	0.002	25	TRUE
32	b2601	b2329	0.624669	0.009683	0.002083	24	TRUE
33	b0754	b3389	0.623606	0.009844	0.002174	23	TRUE
34	b2601	b0754	0.601668	0.01367	0.002273	22	TRUE
35	b2601	b1704	0.567844	0.02176	0.002381	21	TRUE
36	b1704	b3389	0.550411	0.02716	0.0025	20	TRUE
37	b1692	b3281	0.466123	0.06878	0.002632	19	TRUE
38	b3281	b3389	0.448003	0.08181	0.002778	18	TRUE
39	b3281	b1693	0.407678	0.117	0.002941	17	TRUE
40	b3281	b0908	0.387928	0.1376	0.003125	16	TRUE
41	b1704	b3281	0.378799	0.1479	0.003333	15	TRUE
42	b0754	b3281	0.347885	0.1867	0.003571	14	TRUE
43	b1704	b1692	0.326187	0.2176	0.003846	13	TRUE
44	b2601	b3281	0.310293	0.2421	0.004167	12	TRUE
45	b3281	b2329	0.298790	0.261	0.004545	11	TRUE
46	b3281	b0388	0.212592	0.4292	0.005	10	TRUE
47	b0754	b1692	0.203752	0.4491	0.005556	9	TRUE
48	b3281	b3390	0.196603	0.4655	0.00625	8	TRUE
49	b1692	b2329	0.154616	0.5675	0.007143	7	TRUE
50	b1692	b1693	0.145032	0.592	0.008333	6	TRUE
51	b1692	b0908	0.114380	0.6732	0.01	5	TRUE
52	b1692	b3389	-0.100095	0.7123	0.0125	4	TRUE
53	b1692	b3390	-0.027475	0.9195	0.016667	3	TRUE
54	b1692	b0388	0.021264	0.9377	0.025	2	TRUE
55	b2601	b1692	-0.020840	0.9389	0.05	1	TRUE

Appendix D

Metabolic models used in the present study

Small scale metabolic models:

1) Calvin cycle model (Poolman *et al.*, 2003)
2) Potato tuber model (Assmus, 2005)
3) Lactic acid production model (Poolman *et al.*, 2004b)
4) Clavulanic acid production model (Bonde *et al.*, unpublished data)
5) *S. erythraea* metabolic model (Harshil Patel, CSM lab, work in progress and unpublished data)

Genomic scale metabolic models

1) Ecoli-1 model (Fell & Wagner, 2000)
2) E.coli-3 model (Reed *et al.*, 2003)
3) E.coli-4 model (Bonde, Unpublished data)

The metabolic model (ScrumPy file format) data of the above models are included in the supplementary data section on the CDROM media.

For few small metabolic models, the metabolic charts are also drawn and added to the CDROM media.

Appendix E

PyDamage module for damage analysis

This section gives the Python code written for the present study. It is impossible to produce all the pieces of code written during the present study, therefore main modules such as PyDamage is reported in this appendix while all other Python code can be found on the CD-ROM media attached to the dissertation.

Damage analysis module:

```
###################
## Damage Module
## Bhushan Bonde
## March 05, 2004
###################

from ScrumPy.Util import Set, Seq
import ScrumPy

#####import ScrumPy.Util.Set as Set

def damagedic(m):
    """pre:m is spy model
    post: returns three dics, GTDic, NSDic, ELDic"""
    GTDic=GTDamRvR(m)
    NSDic=NSDamRvR(m)
    ELDic=ELDamRvR(m)

    return GTDic, NSDic, ELDic

def GTDamRvR(m):
     '''Pre: m is ScrumPy model object
      Post: Returns a dictionary key: reaction, val: damaged
      reactions'''

     GTDic={}
     for reac in m.sm.cnames:
         GTDic[reac]=[ ]
         rxdam = listRxDamage(m,reac)
         for damagedrx in rxdam:
             GTDic[reac].append(damagedrx)
     return GTDic            ### returns a dictionary

def GTDamRvM(m):
    '''pre: m is a valid ScrumPy Model
      post: returns a dictionary; key:reaction, values:
      metabolites damaged'''

    GTDic={}
    for reac in m.sm.cnames:
        GTDic[reac]=[ ]
        dammetlist = listMetDamage(m,reac)
```

```
        for damagedmet in dammetlist:
            GTDic[reac].append(damagedmet)

    return GTDic        ### returns a dictionary

def listRxDamage(m,reac):
     '''Pre: m: spy model, reac is name of reaction to be
     deleted
     post: returns list of Reactions to be deleted'''

     sm = m.sm.Copy(Conv=float)
     Netd(sm,[reac])

     rxdam = [ ]          # rxdam is list of reactions damaged
     for Reac in m.sm.cnames:
          if Reac not in sm.cnames:
               rxdam.append(Reac)
     return rxdam

def listMetDamage(m,reac):
    '''Pre: m is Valid ScrumPy Model, reac is name of reaction
    to be deleted
    post: returns list of metabolies to be deleted'''

    sm = m.sm.Copy(Conv=float)
    Netd(sm,[reac])
    metdam = [ ]     # metdam is list of metabolites damaged
    for met in m.sm.rnames:
         if met not in sm.rnames:
              metdam.append(met)
    return metdam

def Netd(sm,  reacts):
    """ Pre: sm is valid Sto Mat, reacts is a list of reactions
    Post: none"""
    DelReacts = []
    for r in reacts:
        if r in sm.cnames:
            for met in findMet2Del(sm, r):
                UpdateReac2Del(sm,met,r,DelReacts)
                sm.DelRow(met)
            sm.DelCol(r)
    if len(DelReacts) >0:
         Netd(sm,DelReacts)

def findMet2Del(sm, React):
     """pre: sm is a stoichiometry matrix, React is a list of
     reaction to be deleted
     post: list of metabolites that are products of irreversible
     React, or substrates and products of reversible React"""

     rv = []
     if React in sm.cnames:
          mets = sm.InvolvedWith(React)
          if mets != None:
               for met in mets.keys():
                    Prods = sm.InvolvedWith(met)
                    del Prods[React]
```

```
                    nProds = 0
                    for p in Prods.keys():
                         coeff = Prods[p]
                         if coeff >0:
                              nProds +=1
                         elif p not in sm.Irrevs:
                              nProds +=1
                    if nProds == 0:
                         rv.append(met)

     return rv     ### return list of metabolites to be deleted

def UpdateReac2Del(sm, met,Ignore,DelList):
    """pre: sm is stomat and met is metabolite name 2 be
    deleted, ignore is list to be ingored, DelList is List of
    react
    post: none"""

    reacs = sm.InvolvedWith(met)
    if reacs != None:
        for reac in reacs.keys():
            if not reac in DelList and reac != Ignore:
                   DelList.append(reac)

def NSDamRvR(m):
    '''Pre: m is a Valid ScrumPy Model
     Post: Returns a dictionary: key: reac, values: reactions
damaged'''

    NSDic={}
    smnew = m.sm.Copy(Conv=float)
    DelInitialDeadReacs(m, smnew)

##### done only ones initially to delete original dead reactions
### smnew is the main STOICHIOMETRY matrix from now onwards ####

    for reac in smnew.cnames:
        NSDic[reac]=[]
        dam = Damage(m, smnew, reac)
        if reac not in dam:
            dam.append(reac)                #### self damage
considered
        for damagedrx in dam:
            NSDic[reac].append(damagedrx)

    return NSDic        ### return nullspace damage dictionary

def Damage(m, sm, r):
    '''Pre: sm is a StoMat, r is a reaction to be deleted
    post: returns list of dead reactions'''
    sm = sm.Copy()
    sm.DelCol(r)
    dead = GetDeadReacs(m, sm)
    dead.append(r)

    return dead

def DelInitialDeadReacs(m, sm):
```

```
    '''Pre: sm is a valid StoMatrix
    post: Delete the dead reactions form Sto Matrix'''

    DeadReacs = []
    enzsubset=m.EnzSubsets()
    deadset=enzsubset.DeadSSs()

    if len(deadset) > 0:
        for rx in enzsubset[deadset[0]]:
            DeadReacs.append(rx[0])
        for dreac in DeadReacs:
            sm.DelCol(dreac)

def GetTransportRx(m):
    return m.Transporters(True)

def GetDeadReacs(m, sm,Thresh=1e-10):
    '''Pre:sm is a valid StoMatrix
    post: returns list of Dead reactions'''

    rv = []
    ker = sm.NullSpace()

    if ker == None:
        rv = sm.cnames[:]
    else:
        for r in ker.rnames:
            if ScrumPy.Util.Seq.AllMatch(ker[r], lambda x,
thr=Thresh: abs(x) <= thr):
                rv.append(r)

        intercycles=GetInternalCycles(m, ker)

        for k in intercycles:
            for reac in intercycles[k]:
                if reac not in rv:
                    rv.append(reac)

    return rv

def GetInternalCycles(m, null):
    """Pre: m is spy model, null is Nullspace
    post: retuns a dic of Interal Cycles in a model"""
    intcycdic={}
    TransRx=m.Transporters(True)
    for cname in null.cnames:
        col=null.GetCol(cname)
        rxls=[ ]
        for rx in null.rnames:
            if null[rx, cname] != 0:
                rxls.append(rx)
        if not Set.DoesIntersect(rxls, TransRx):
            intcycdic[cname]=rxls
    return intcycdic

def ELDamRvR(m):
    """ Pre: m is valid ScrumPy model object
    Post : Returns a dictinary: key: reaction
```

```
            Values : reactions damaged """

    ELDic={}

    elm=getvalidmode(m)

    for reac in m.sm.cnames:
        ELDic[reac]=[ ]
        for damagedrx in Damager(elm, reac):
            ELDic[reac].append(damagedrx)

    return ELDic

def getvalidmode(m):
    """ pre: m->model,
     post: returns only valid modes (removes futile modes)"""

    elm=m.ElModes()
    fut=elm.Futile()
    if len(fut) > 0:
        for mode in fut.mo.rnames:
            elm.mo.DelRow(mode)
            elm.sto.DelRow(mode)
    return elm

def Damager(modes, reac):
    """pre: modes=m.Elmodes, reac is reaction to be damaged
    (removed) from model
    post: returns list of damaged reac => reactions from those
    ELM which don't use flux through reac"""

    rmodes = modes.NoFlux(reac)   ## modes which do not pass
through 'reac'
    notdamrx=[]                ## list of undamaged rx as they
                              ### belong to not passing elmodes
    damrx=[]                  ## list of damaged reactions
    damrx.append(reac)           ##  self damage

    for elm in rmodes.mo.rnames:
        for rx in rmodes.mo.cnames:
            if rmodes.mo[elm,rx] != 0:
                if rx not in notdamrx:
                    notdamrx.append(rx)

    ELMviaReac=Set.Complement(modes.mo.rnames, rmodes.mo.rnames)
                                    ### list of modes
    for elm in ELMviaReac:
        for rx in modes.mo.cnames:
            if modes.mo[elm, rx] != 0:
                if rx not in damrx:
                    damrx.append(rx)
    rv=[]
    for rx in damrx:
        if rx not in notdamrx:
            rv.append(rx)
    return rv
```

Appendix F

List of publications, conference oral and poster presentations originated from the present study

List of publications

1 Poolman, M. G., Bonde, B. K., Gevorgyan, A. *et al.* (2006). Challenges to be faced in the reconstruction of metabolic networks from public databases, IEE Systems Biology, (in press, preprint attached)

Conference/workshop oral presentations

1. Bonde, B., Structural Modelling of *E.coli* metabolic network, 2004, at Systems Biology Dynamics: from Genes to Organisms, organised by Centre for Nonlinear Dynamics, McGill University, Montreal, Canada.
2. Bonde, B., *In-silico* structural modelling of metabolic network, 2004, at Postgraduate Research Students Seminar, Oxford Brookes University, Oxford.
3. Bonde, B., Poolman, M., Fell, D., Structural Analysis of Genomic Scale Metabolic Models, 2004 at Young-BTK workshop 2004, Oxford.
4. Bonde, B., Modelling of large genomic scale metabolic networks, 2003, at School of Biological and Molecular Sciences Research Symposium, Oxford.
5. Fell D. A., Bonde B., Poolman M. G, The substructure of large metabolic networks, 2005, *FEBS Journal* 272 (s1), [H4-003] at FEBS conference.

Poster presentations

1. UK - Young Bioinformatics Forum, Institute of Physics, London, 21 Oct 2005
2. Microbial Systems Biology workshop at University of Surrey, 14-15 Jul 2005.
3. Mathematical Challenges in Systems Biology workshop at Univ. of Warwick, 27-29 Oct 2004.
4. UK - Young Bioinformatics Forum, Said Business School, Oxford, 20 Oct 2004.
5. BioThermoKinetics (BTK) meeting Oxford, 3-6 Sep 2004.
6. Bioscience 2004 conference at Glasgow, UK, 18-22 Jun 2004.
7. Molecules as Modulators: Systems biology challenges in chemistry workshop at Aventis Pharmaceuticals, Frankfurt, Germany, 29-31 Jan 2004.