The Structural Analysis Of Metabolism On A Genomic Scale

Harshil Patel

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

Cell Systems Modelling Group
School of Life Sciences

OXFORD
BROOKES
UNIVERSITY

October 2008

Abstract

A minimal structural model encompassing the flow of carbon through the central metabolism of *Saccharopolyspora erythraea* was built and used to assess the application of hierarchical clustering techniques for the grouping and, subsequent interpretation of elementary modes. The results indicate that clustering techniques may prove to be a valuable tool for the functional characterisation and grouping of the potentially large datasets yielded by elementary modes analysis, whether clustered by their reaction usage or net stoichiometry. In another study, clustering methods were also used for the reconstruction of phylogenetic trees based on the enzyme complement of 369 prokaryotes. A comparison of the resulting phylogenetic trees with 16S rRNA-based trees indicated some interesting phenotypic and taxonomic discrepancies. The availability of more reliable database information will help elucidate the positions of both the anomalous and well-defined organisms in the enzyme-based trees. Genome-scale metabolic networks created in such an automated manner are far from being complete and require verification and refinement before they are suitable for modelling purposes. An investigation was carried out as part of a group effort to address the problems encountered therein, and to identify potential steps that can be taken to improve the quality of the model in order to generate more reliable phenotypic predictions. Taken together, the investigations reported here indicate that co-ordinated research efforts at the systems-level may need to be reevaluated in order to increase the potential knowledge that can be gained from building and interpreting genome-scale models of metabolism.

Dedication

To my loving Parents,

Your sacrifices have made our dreams come true.

Acknowledgements

This PhD project could not have been possible without the united attention that I was offered from my supervisor's Prof. David Fell and Dr. Mark Poolman. They were always there for me when I needed them and in my opinion were 'model' mentors. I was always given a chance to express myself and my ideas throughout the course of the project, and was thereafter guided in the right direction. David lived up to his reputation as the 'human computer' and his passion for our field of study, general advice and patience were invaluable to me. He allowed me to demonstrate alongside him in undergraduate biochemistry practicals and I loved the experience. Excluding biological insights, Mark greatly helped my programming and computer-based development and I cannot thank him enough. He always made himself available to me for short discussions in the lab and to provide me with the information that I needed help with, and more. I would also like to thank my colleagues, Bhushan Bonde, Albert Gevorgyan and Unni, as fellow students on the same doctoral boat. We shared our ideas and helped motivate each other to get through the ups and downs that are inevitable at this level of study.

The administrative and supplementary parts of the course were very well handled by the postgraduate staff in the School of Life Sciences. In particular, I would like to acknowledge Farida, Jill Organ, David Evans and Jill Helmsley. Amongst the development programmes that were on offer through the department, the Biotechnology Yes competition had the most pronounced impact on my scientific outlook. Aside from the more familiar biotechnology component of the competition, I gained some important business acumen with regard to startup companies. I would also like to express my gratitude to the Biotechnology and Biological Sciences Research Council for providing me with the funding for writing this dissertation, and sponsoring my attendance at the UK GRAD School in the Lake District.

My friends gave me the opportunity to get away from the project just when I needed it, and to return with replenished ideas in a not so vegetative state. My parents and sister have supported me throughout the my life and I owe this work to them. Over the years they have helped and supported me to develop a mentality that could overcome the challenges a PhD has to offer, the fruits of which are evident in its successful completion.

Thank you all once again.

Contents

List of Abbreviations

Metabolite Abbreviations

ACCOA	Acetyl-CoA
ADP	Adenosine Diphosphate
AMP	Adenosine Monophosphate
ATP	Adenosine Triphosphate
BPG	D-Glycerate 1,3-Bisphosphate
CIT	Citrate
DHAP	Dihydroxyacetone Phosphate
E4P	D-Erythrose 4-Phosphate
F6P	D-Fructose-6-Phosphate
FAD/FADH$_2$	Flavin Adenine Nucleotide (oxidized/reduced)
FBP	D-Fructose 1,6-Bisphosphate
FUM	Fumarate
G1P	D-Glucose-1-Phosphate
G6P	D-Glucose-6-Phosphate
GAP	D-Glyceraldehyde 3-Phosphate
GLC	D-Glucose
GLX	Glyoxylate
ICIT	Isocitrate
KDPG	2-Keto-3-Deoxy-6-Phosphogluconate
MAL	Malate
NAD/NADH	Nicotinamide Adenine Dinucleotide (oxidized/reduced)
NADP/NADPH	Nicotinamide Adenine Dinucleotide Phosphate (oxidized/reduced)
OXOA	Oxaloacetate
OXOG	2-Oxoglutarate
P2G	D-Glycerate 2-Phosphate
P3G	D-Glycerate 3-Phosphate
PEP	Phosphoenolpyruvate
PGC	6-Phospho-D-Gluconate
PGL	6-Phospho-D-Glucono-1,5-Lactone
P_i	Inorganic Phosphate
PP_i	Inorganic Pyrophosphate
PYR	Pyruvate
R5P	D-Ribose 5-Phosphate
RU5P	D-Ribulose 5-Phosphate
SUCC	Succinate
SUCCOA	Succinyl-CoA
S7P	D-Sedoheptulose 7-Phosphate
X5P	D-Xylulose 5-Phosphate

Other Abbreviations

ASCII	American Standard Code for Information Interchange
BLAST	Basic Local Alignment Search Tool
COG	Clusters of Orthologous Groups
DDBJ	DNA Databank of Japan
DOE	Department of Energy
EBI	European Bioinformatics Institute
EC	Enzyme Comission
EM	Elementary Mode
EMA	Elementary Modes Analysis
EP	Extreme Pathway
EPA	Extreme Pathway Analysis
ExPASy	Expert Protein Analysis System
FASTA	Fast All
FBA	Flux Balance Analysis
HGT	Horizontal Gene Transfer
KEGG	Kyoto Encyclopedia of Genes and Genomes
LP	Linear Programming
MEGA	Molecular Evolutionary Genetics Analysis
MFA	Metabolic Flux Analysis
MILP	Mixed-Integer Linear Programming
MOMA	Minimization of Metabolic Adjustment
MUSCLE	Mltiple Sequence Comparison by Log-Expectation
NCBI	National Centre for Biotechnology Information
NMR	Nuclear Magnetic Resonance
ORF	Open Reading Frame
PAUP	Phylogenetic Analysis Using Parsimony
RBC	Red Blood Cell
SBML	Systems Biology Markup Language
SWIG	Simplified Wrapper & Interface Generator
TCA	Tricarboxylic Acid Cycle
UPGMA	Unweighted Pair Group Method Using Arithmetic Averages
VGT	Vertical Gene Transfer

List of Urls

Biological Databases

BioCyc	`http://www.biocyc.org`
BioCyc Yeast Maps	`http://pathway.yeastgenome.org/biocyc`
BLOCKS	`http://blocks.fhcrc.org`
Brenda	`http://www.brenda-enzymes.info`
Celera Genomics	`http://www.celera.com`
COG	`http://www.ncbi.nlm.nih.gov/COG`
DDBJ	`http://www.ddbj.nig.ac.jp`
EBI	`http://www.ebi.ac.uk`
EcoCyc	`http://www.ecocyc.org`
ENZYME	`http://www.expasy.ch/enzyme`
ExPASy	`http://www.expasy.ch`
IntEnz	`http://www.ebi.ac.uk/intenz`
InterPro	`http://www.ebi.ac.uk/interpro`
Joint Genomes Institute	`http://img.jgi.doe.gov/cgi-bin/pub/main.cgi`
KEGG	`http://www.genome.jp/kegg`
KEGG GENES Ftp Site	`ftp://ftp.genome.jp/pub/kegg/genes/organisms`
MetaCyc	`http://www.metacyc.org`
NCBI	`http://www.ncbi.nlm.nih.gov`
NCBI Taxonomy Website	`http://www.ncbi.nlm.nih.gov/Taxonomy`
Pathway Resource List	`http://www.cbio.mskcc.org/prl`
PROSITE	`http://www.expasy.ch/prosite`
Reactome	`http://www.reactome.org`
Saccharomyces Genome	`http://www.yeastgenome.org`

Software Tools

BLAST	`http://www.ncbi.nlm.nih.gov/blast/Blast.cgi`
CellNetAnalyzer	`http://www.mpi-magdeburg.mpg.de/projects/cna`
CLUSTALW	`http://www.ebi.ac.uk/Tools/clustalw2`
Copasi	`http://www.copasi.org`
EBI MUSCLE Server	`http://www.ebi.ac.uk/Tools/muscle/index.html`
FASTA	`http://www.ebi.ac.uk/fasta33`
Gepasi	`http://www.gepasi.org`
Jarnac	`http://www.sys-bio.org/software/jarnac.htm`
MEGA	`http://www.megasoftware.net`
metaSHARK	`http://bmbpcu36.leeds.ac.uk/shark`
Metatool	`http://pinguin.biologie.uni-jena.de/bioinformatik/networks/index.html`
MKDOM	`http://prodom.prabi.fr/prodom/xdom`
MUSCLE	`http://www.drive5.com/muscle`
NJPlot	`http://pbil.univ-lyon1.fr/software/njplot.html`
Pathway Tools	`http://bioinformatics.ai.sri.com/ptools`
PAUP	`http://paup.csit.fsu.edu`

PHYLIP	`http://evolution.genetics.washington.edu/phylip.html`
PRIAM	`http://bioinfo.genotoul.fr/priam`
PSI-BLAST	`http://blast.ncbi.nlm.nih.gov/Blast.cgi`
PySCeS	`http://pysces.sourceforge.net`
SEED	`http://www.theseed.org`
SCAMP	`http://www.sys-bio.org`
ScrumPy	`http://mudshark.brookes.ac.uk/ScrumPy`
SplitsTree	`http://www.splitstree.org`
T-COFFEE	`http://www.tcoffee.org`
TREEDIST	`http://evolution.genetics.washington.edu/phylip/doc/treedist.html`
YANA	`http://www.biozentrum.uni-wuerzburg.de/yana.html`

Programming Resources

BioPython	`http://www.biopython.org`
C Language	`http://www.cprogramming.com`
Java Language	`http://java.sun.com`
MATLAB	`http://www.mathworks.com`
Python IDLE Interface	`http://www.python.org/idle`
Python Language	`http://www.python.org`
SBML	`http://www.sbml.org`
SciPy	`http://www.scipy.org`
SWIG	`http://www.swig.org`

CHAPTER 1

Metabolic Modelling

1.1 Introduction

Mathematical modelling is an important research area in biology and the relatively newer field of bioinformatics[1]. The representation of a metabolic network in a mathematical form is called a metabolic model. Once built, such a model can be analysed to characterise the properties of the system as a whole or for computing the theoretical limits of the systems capabilities. Furthermore, metabolic modelling techniques can be utilised for hypothesis generating and testing, driving both *in silico* and *in vivo* experimentation. The two most commonly used approaches to metabolic modelling are kinetic and structural modelling:

- **Kinetic modelling**
  Represents a quantitative approach to metabolic modelling. Such a model can be used to describe the time-dependent changes of the variables of the system (e.g. metabolite concentrations, reaction rates) when the experimental system is perturbed [1]. Therefore, along with reaction stoichiometries, further information is also required as to the kinetic parameters (e.g. K_m and V_{max}) for each enzymatic reaction. Kinetic modelling is the method of choice for small hand-built models in which kinetic parameters have been, or are, possible to quantify. However, it is practically impossible to build a kinetic model for genome-scale[2] models since the availability of specific enzyme reaction rates is one of the major limiting factors. Additionally, due to model size, the ensuing analysis becomes computationally intensive and, without the necessary theoretical and computational tools, the results obtained from the analysis will be difficult to interpret. A number of kinetic models for various metabolic systems as well as for membrane transport processes have been investigated [1] (see Section 1.4 for an example).

[1] the collection, organisation and analysis of large amounts of biological data, using networks of computers and databases.

[2] in this context, all of the metabolic reactions that could be determined to take place in an organism based on genome annotation and biochemical literature.

- **Structural modelling**
  Whereas the aim of kinetic modelling is to predict the system properties on the basis of the knowledge of the network topology[3] and the kinetic parameters of enzymes, structural modelling only requires the former information as input data. Analysis of the structure of the network requires mainly reaction stoichiometries, which are often well-known. This implies that for the most part structural models are reconstructed based on the biochemical reactions assumed to be present within the network. The exclusion of kinetic data restricts structural models in terms of the level of predictions that can be made for a given system. Nevertheless, this is in some ways compensated by the ability to build larger structural models from which useful conclusions can be drawn regarding the invariant properties of the network.

The investigations described in this dissertation have solely employed structural modelling techniques, owing to their simplicity and scope of use for larger genome-scale models.

1.2 Data requirements

A structural metabolic model typically consists of a list of biochemical reactions and their associated metabolites. In order to reflect the characteristics of the system under interrogation certain rules apply when defining these entities for modelling purposes [2].

1.2.1 Metabolites

When carrying out any modelling investigation two types of metabolite[4] may be defined; external and internal. Those metabolites that flow across the system boundary and are made available to the system as buffers are termed external metabolites. Externality is determined by:

- source or sink metabolites which are consumed (e.g. glucose) and/or produced (e.g. ethanol) by the system to mimic media conditions.
- metabolites that are likely to be in constant exchange with the extracellular environment in living cells (e.g. water, oxygen and carbon dioxide).

[3] the most basic feature of any network used to describe the pattern of interactions between its components.

[4] a substance which participates in a biochemical reaction and represent the intermediates and products of metabolism.

Internal metabolites are defined within the model as those which are likely to be generated and utilised as part of the intracellular metabolism of the system. In some cases, the decision to make a particular metabolite internal or external depends more on the modelling investigation than the ability to reflect the genuine properties of the system. For example, external definitions may be expanded to include:

- metabolites that are highly connected (i.e. used by many reactions) within the model definition (e.g. ATP/ADP and NADP$^+$/NADPH). Depending on the modelling objectives these 'hub' metabolites [3] may be made external to reduce the connectivity within the model and, as discussed later (Section 1.6.3.1) to reduce the computational issues that arise during the analysis of the model [4, 5].
- polymeric metabolites (e.g. starch and DNA) since the reaction stoichiometry does not imply the net amount of monomers incorporated into the polymer.

1.2.2 Reactions

A biochemical reaction can be defined as a process in which one or more reactants interact and produce one or more products, usually catalysed by an enzyme. Reactions that can proceed without the need for an enzyme are termed spontaneous. Transport reactions are those reactions that bring about for the movement of metabolites between cellular compartments and are not necessarily mediated by enzymic conversions or facilitated diffusion (i.e. can also occur by concentration gradients).

The reaction data for metabolic modelling includes a reaction equation, its corresponding directionality (i.e. reading the reaction from left to right or vice versa) and reversibility criteria (i.e. whether the reaction can be catalysed in both directions or not). Additional information regarding the cellular compartmentation of the system metabolites is also essential. Integration of this information into the model definition will allow the system to reflect *in vivo* conditions as accurately as possible.

1.3 Data resources

The reductionist approach to biology, which over the years has generated information about individual cellular components and their functions, is now being accelerated by the emergence of genomics[5]. The advent of comprehensive measurement technologies yield large-scale datasets on many cellular components and their interactions [6]. As a consequence, there has been a vast expansion in academic (e.g. European Bioinformatics Institute† and National Center for Biotechnology Information†; note that † superscript will be used throughout the thesis to direct the reader to the List of URLs section) and commercial (e.g. Celera genomics†) organisations which provide the bioinformatics infrastructure necessary to tabulate, curate, and retrieve the required data [7]. Visualisation tools and statistical analysis methods for data analysis are also becoming available within these frameworks.

Large-scale sequencing projects have not only provided complete sequence information for a number of genomes, but also allowed the development of integrated pathway-genome online databases that provide organism-specific connectivity maps of metabolic and, to a lesser extent, other cellular networks (see Pathway Resource List†). Several databases are available to reconstruct a metabolic network from genome information (Section 4.2), these may be of two types:

- **General-purpose**
  The KEGG† [8], MetaCyc† [9] and Reactome† [10] databases are the most popular in this category and contain sequence data for a large spectrum of organisms. A variety of additional information on genes, enzymes, proteins, and ligands is also included.

- **Organism-specific**
  Include the EcoCyc† [11] and *Saccharomyces* Genome Database† [12] for *Escherichia coli* and yeast, respectively. Databases of this type are used to provide a user-friendly interface for the access and inspection of the metabolic characteristics (i.e. experimental and sequence data) of a single genome. Additional organism-specific information such as viability of mutants and availability of clones may also be available.

While the recent dramatic increase in the number of pathway databases is beneficial for biologists, it also presents several important challenges. Existing databases are very heterogeneous; data can be incomplete, inconsistent or approximate (i.e. annotating by homology as opposed to experimentally). This multiplicity of information sources can be overwhelming for researchers who simply wish to find information about genes or pathways of interest in a standardised fashion. During the course of this project it has become apparent that database-derived information has many discrepancies when used to create large metabolic models (Section 4.5).

[5] the study of an organism's entire genome.

1.4 Hand built vs. genome scale models

Hand-built metabolic models are very useful for studying small structural networks (e.g. glycolysis) or for kinetic investigations where reaction parameters are also essential. Models built in this manner tend to be a very precise reflection of the system under study since the size of the system permits the individual error screening of all the reactions before and after their inclusion in the model. The first whole-cell model was developed in the late 1980's for the human red blood cell (RBC) in order to simulate its kinetic behaviour [13]. The small RBC metabolic network consists of four well-documented pathways (i.e. glycolysis, the pentose pathway, adenosine nucleotide metabolism, and the Rapoport-Luebering shunt). The model has continually been updated [14, 15, 16] and has, for example, been investigated to simulate common RBC pathologies stemming from hereditary glucose-6-phosphate dehydrogenase deficiency [17].

The increasingly thorough genome-sequencing and annotation efforts currently being undertaken permit the reconstruction of organism-specific biochemical networks of metabolism. When building models of this size (i.e. typically in excess of 250 reactions) the precision attributed to small hand-built models is in some ways compensated for by the ability to investigate a substantial fraction of the reactome[6] of the organism. To date, a number of genome-scale networks have been derived from annotated genome data [18]. Furthermore, mathematical models and their computer simulation allow us to examine the integrated function of the reconstructed metabolic network [19]. Prokaryote genome-scale reconstructions such as those for *E. coli* [20] and *Streptomyces coelicolor* [21] have been used to predict cellular behaviour under different physiological conditions. When compared to other cell types, the whole-cell RBC network reconstruction has proved to be the most fruitful metabolic model created thus far. Therefore, aside from its usefulness as a kinetic model it has also been used as a model system to examine and validate structural analysis procedures [22]. Structural modelling techniques can be a very useful tool as the first step towards cataloguing and characterising an organism in terms of its metabolic properties, and will be discussed further in Section 1.6 and Chapter 4.

1.5 Structural modelling: theory

The following sections will aim to elaborate on the level of theoretical understanding that is required for the investigation of structural metabolic models.

[6] the entire reaction complement to be found within a biological sample, such as a single organism.

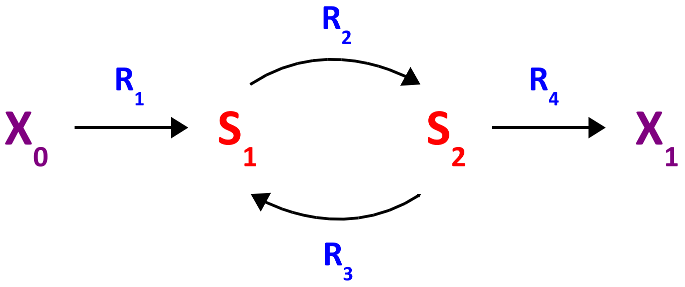


Figure 1.1 – A simple network consisting of 4 reactions (R_1–R_4), 2 internal metabolites (S_1 and S_2) and 2 external metabolites (X_0 and X_1).

1.5.1 Stoichiometry matrix

Under a given set of conditions, the stoichiometry matrix (**N**) is a compact mathematical representation of a biochemical network. The stoichiometry matrix for the system in Figure 1.1 is:

$$\mathbf{N} = \begin{pmatrix} 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & -1 \end{pmatrix} \tag{1.1}$$

where the dimensions $m \times n$ of **N** are such that m (number of rows) is equal to the total number of metabolites[7] and n (number of columns) is equal to the total number of reactions. Each element in **N** is called a stoichiometric coefficient and it indicates:

- whether a metabolite takes part in a particular reaction or not.
- the number of molecules of metabolite participating in that reaction.
- whether it is a reactant or product, according to the sign of the element.

Stoichiometric network reconstructions and their ensuing mathematical analyses can be used further to determine the total potential of an organisms reaction network (reviewed in [1]) (Section 1.6). Analytical methods based on the direct interrogation of **N** can be employed for a purely stoichiometric analysis and are described further in Section 1.6.1.

1.5.2 The steady state concept

The mass balance of a metabolite is defined as the difference between its rate(s) of production and the rate(s) of consumption. Consider the single metabolite system shown in Figure 1.2, the rate of change of internal metabolite, S, is given by:

[7] unless stated otherwise **N** only includes the stoichiometry of internal metabolites.

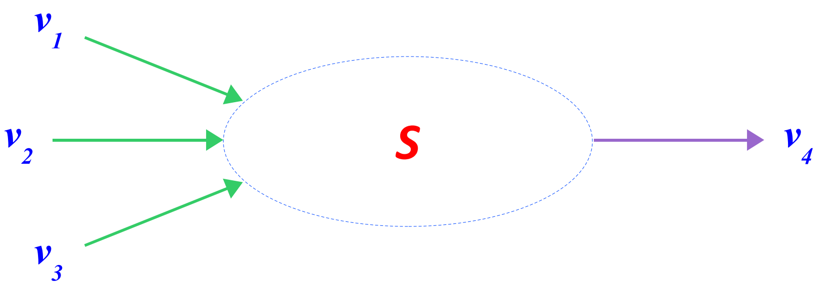


Figure 1.2 – Illustration to demonstrate the steady state concept.

$$\frac{dS}{dt} = v_1 + v_2 + v_3 - v_4 \tag{1.2}$$

Alternatively, employing basic linear algebra, Equation 1.2 can be expressed as a product of the stoichiometry matrix and a column vector of reaction rates (**v**):

$$\frac{dS}{dt} = \begin{pmatrix} 1 & 1 & 1 & -1 \end{pmatrix} . \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} \tag{1.3}$$

The steady state assumption allows the modeller to equate the total rate of production for any internal metabolite to the total rate of its consumption (i.e. indicating that the macroscopic variables - flux and metabolite concentrations - change only to a tolerable extent over a specific time span) [1, 23]. Taking this into account, the three inputs to the system in Figure 1.2 must equal the total output in order to keep [S] constant. Therefore, Equation 1.3 becomes:

$$\frac{dS}{dt} = \begin{pmatrix} 1 & 1 & 1 & -1 \end{pmatrix} . \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = 0 \tag{1.4}$$

Furthermore, Equation 1.4 can be summarised as:

$$\mathbf{N}.\mathbf{v} = 0 \tag{1.5}$$

where for any given system **N** describes the network topology and can be considered as a constant, and **v** represents its variables.

1.5.3 Null space matrix

Structural modelling does not require any kinetic information as represented by the vector of reaction rates in Equation 1.5. Thus, by taking the stoichiometric matrix and steady state conditions into account, Equation 1.5 can be solved in order to estimate the possible flux distributions of the system. One valid solution is $\mathbf{v} = \mathbf{0}$, however, this is not very informative since it identifies a state within which all the reactions in the system are incapable of carrying flux. Alternatively, linear algebraic methods (represented by a set of homogeneous equations[8]) may be utilised to calculate the subspace of all possible solutions to Equation 1.5. This subspace is called the null space of $\mathbf{N}$ and can be described mathematically by a kernel matrix $\mathbf{K}$[9] [1], whose columns are linearly independent[10] vectors spanning this subspace (i.e. they form a basis) [24], to give the equation:

$$\mathbf{N}.\mathbf{K} = \mathbf{0} \tag{1.6}$$

where each column of $\mathbf{K}$ is a possible solution to Equation 1.5 and each row represents a single reaction. The dimension of $\mathbf{K}$ (i.e. number of linear basis vectors) is the difference between the number of metabolites (m) and the rank of $\mathbf{N}$ (rank($\mathbf{N}$)). If m is equal to rank($\mathbf{N}$), there may be a unique solution to the system of equations, otherwise, it has infinitely many solutions.

Gaussian elimination [1] or singular value decomposition [25] methods may be used to determine the null space. The former method is easier to understand and to implement algorithmically whereas the latter is more complex but suited for larger stoichiometry matrices [26]. Using these algorithms, multiple instances of $\mathbf{K}$ may be obtained which satisfy Equation 1.5, indicating that the basis of the null space is not unique. Using Figure 1.1 as an example, two possible instances of $\mathbf{K}$ are:

$$\mathbf{K}_1 = \begin{array}{c} R_1 \\ R_2 \\ R_3 \\ R_4 \end{array} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \mathbf{K}_2 = \begin{array}{c} R_1 \\ R_2 \\ R_3 \\ R_4 \end{array} \begin{pmatrix} -1 & 0 \\ -1 & -1 \\ 0 & -1 \\ -1 & 0 \end{pmatrix} \tag{1.7}$$

All the reactions in Figure 1.1 have been defined as irreversible. As mentioned earlier, the columns of $\mathbf{K}$ can be interpreted as potential flux distributions through the network. This implies that the first column of $\mathbf{K}_1$ is the forward conversion

[8] set of linear equations without a constant term whereby right hand side is equal to zero.

[9] unless stated otherwise, $\mathbf{K}$ will represent the (right) null space of $\mathbf{N}$ henceforth in the dissertation.

[10] none of the vectors in $\mathbf{K}$ can be written as a linear combination of finitely many other vectors in the collection.

of X_0 to X_1 via R_2 and the second is the cyclic flux between R_2 and R_3. The opposite scenario is apparent in $\mathbf{K}_2$, and although, it is a valid set of solutions to the system, it violates the irreversibility criteria that has been laid down for the network. As discussed more extensively in the next section, the constraint-based approaches to structural analysis, mainly developed by Palsson and co-workers [27] take reaction irreversibility into account by imposing a non-negativity constraint on those components to determine all the possible routes through the network (Section 1.6.3). Additional constraints such as experimentally determined flux values can also be declared to calculate the remainder of unknown fluxes (Section 1.6.3.3).

1.6 Structural modelling: analysis

Structural models have been investigated using a variety of approaches [1]. As introduced in the following sections there are three general approaches that are considered:

- **Non-steady state analysis**
  The stoichiometry matrix forms the most basic feature of a biochemical network and direct analysis of its components (i.e. how metabolites are connected to each other by reactions) is both trivial and also very useful to identify the underlying properties of the system.

- **Null space analysis**
  Interrogation of the null space of the stoichiometry matrix can be used to identify the characteristics of the system at steady state (e.g. determining whether a reaction can carry flux at steady state).

- **Constraint-based analysis**
  The incorporation of additional constraints into a structural model helps to reduce the possible flux distributions at steady state and thereby limits the range of allowable phenotypes (e.g. imposing a thermodynamic constraint for irreversible reactions).

The section structure described above is not mutually exclusive and certain analytical techniques may be calculated at more than one level within the hierarchy.

1.6.1 Non-steady state analysis

Pure stoichiometric analyses, in the simplest case, is analogous to tracing the route from a substrate to a product in a metabolic pathway from a biochemistry textbook and reporting the number of steps involved. In other words, the static representation of the network is used to draw useful conclusions regarding the inherent pathway structure and organisation of the metabolites in the network.

1.6.1.1 Graph theoretic analysis

Graph theory can be defined as the study of graphs to model the pairwise relations between objects from a certain collection. A graph $G(V, E)$ is a mathematical object that represents a system of elements that interact or regulate each other, where V is the set of nodes (points or vertices), and E is the set of edges (links) connecting pairs of nodes [28]. Graphs may be supplemented by additional information for their constituents:

- Edges may be directed to represent the flow of material from a source (head) to a target (tail), or non-directed for mutual interactions.
- More than one type of node can be represented in multi-partite graphs as explained later.
- Weights, strengths or reaction speeds may be assigned to edges to indicate information from experimental sources [29].

Types of graphical representations The four primary graph representations used for metabolic networks include:

- *Compound graphs*
  Used to model a set of chemical reactions. Nodes represent the chemical compounds and the edges between them represent a shared reaction. The main limitation with this type of graph is its poor descriptive power owing to the fact that it is impossible to distinguish whether any two reactants are involved in the same reaction.
- *Reaction graphs*
  Nodes are reactions and an edge is placed between two nodes if they share at least one chemical compound (i.e. product or substrate). Reversibility information may also be taken into account by employing directed edges.
- *Bipartite graphs*
  Form of multi-partite graph with two classes of node whereby no edges can form a relationship between nodes of the same set. Therefore, for the graph definition $G(V, E)$, V can be divided into two disjoint sets V_1 (reactions) and V_2 (compounds) where $V = V_1 \cup V_2$ and every edge connects a node in

V_1 to one in V_2. The KEGG database employs bipartite graphs for pathway visualisation which contain two types of node (i.e. reactions and metabolites) and edges which represent their interactions.

- *Hypergraphs*
  Forms a generalisation of a compound graph where a edge relates a set of products to a set of substrates. This type of graph can be easily converted into a bipartite graph and vice versa. The MetaCyc database contain pathway representations of this type.

Compound and reaction graphs were employed by Wagner and Fell [30] to carry out a graph theoretical analysis of central routes of energy metabolism and small-molecule building block synthesis in *E. coli*; a selection of the analytical methods used will be introduced later. Jeong *et al.* [31] modelled the metabolic networks of 43 organisms from all three domains of life as bipartite graphs, permitting a comparative analysis showing that they have the same topological scaling properties.

Network measures Once a metabolic network has been reconstructed in terms of its respective interactions, a graph theoretical analysis can be carried out to provide biological insights into the structural organisation and potential functionality of the system. Numerous measures have been defined for this purpose, they include:

- *Degree and degree distribution*
  The degree of a node is simply the number of edges that it is involved with or the number of adjacent nodes. For directed graphs this can be separated into in-degree (number of edges that point to that node) and out-degree (number of edges that start at that node). The degree distribution $P(k)$ of the whole network can also be calculated and indicates the proportion of nodes that have degree k. It can be calculated by counting the number of nodes $N(k)$ with k = 1,2,3... edges and dividing by the total number of nodes [3]. Scale-free metabolic networks are characterised by a degree distribution that follows a power law $P(k) \sim k^{-\gamma}$, where γ is the degree exponent [32]. The value of γ is important for determining the importance of the hubs (highly connected nodes) in the system [3]. In other words, a scale-free network typically contains a few hubs (e.g. ATP) that are involved in numerous reactions and vice versa [30, 31, 33].

- *Shortest path and mean path length*
  The number of edges to get from one node to another in a network is defined

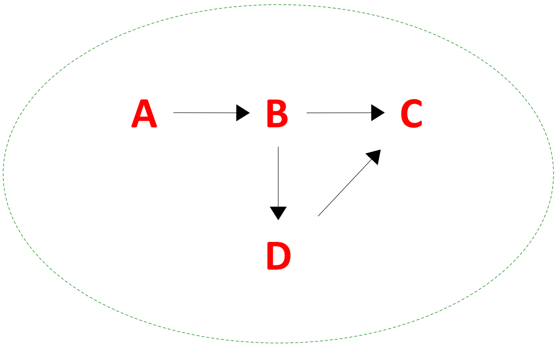


Figure 1.3 – A graphical representation of a metabolic network containing one orphan metabolite (A) and one dead-end metabolite (C).

as the distance or path length [3]. Therefore, the shortest path is the route that has the smallest distance between any given pair of nodes. This measure does not apply for those pairs of nodes between which a route does not exist. When applied to the entire network, the average of shortest path lengths over all pairs of nodes in a network is called the network diameter [32].

1.6.1.2 Orphan and dead-end metabolites

Orphan metabolites[11] are those metabolites that are only produced or consumed by one reaction within the model. Metabolites of this type clearly cannot be balanced and hence reactions involved with them, and quite possibly additional reactions, must be dead[12]. Dead-end metabolites are those that are only produced or consumed by more than one reaction within the model. They pose similar but less serious problems than those incurred by orphan metabolites, since it may be possible to balance a dead-end metabolite if one of the reactions involved with it is reversible. All orphan metabolites are dead-end metabolites but not vice versa (Figure 1.3).

A list of orphan metabolites may be obtained from the stoichiometry matrix by finding those rows with only one positive or negative coefficient. Similarly, dead-end metabolites can be identified by rows with only positive or negative coefficients in the stoichiometry matrix. The issues with orphan and dead-end metabolites become more apparent when building genome-scale metabolic models in an automated fashion, and are discussed further in Chapter 4.

1.6.1.3 Damage analysis

Damage analysis is a method that can be used to investigate the extent of loss induced in a metabolic network by the removal of a single enzyme [34]. The

[11] a term coined within our group.

[12] reactions that cannot carry flux at steady state (Section 1.6.2.1).

analysis is initiated by the removal[13] of all those reactions that are exclusively associated with an enzyme of interest. The ensuing damage analysis can then be used to assign a damage score to the enzyme based on, for example, the number of metabolites whose production the absence of the enzyme prevents. Ultimately, this method may be used as a potential indicator of the importance of an enzyme within the metabolic network.

Lemke *et al.* carried out damage analysis alongside *in vivo* enzyme knockouts to correlate the removal of a particular enzyme with the viability of *E. coli* [34]. Single knockout of 91% of the enzymes caused little damage to the network and the remainder caused serious damage. Experimental validation using the enzymes in the latter set confirmed that these enzymes were essential to the viability of *E. coli*.

1.6.1.4 Conserved moieties

Conserved moieties are those chemical entities (atoms, ions, assemblies of atoms or ions) whose total concentration remains constant in a system, regardless of the kinetics of individual reactions [1]. Typical examples include ATP, ADP and AMP forming a group of metabolites that conserves the adenylate moiety, whereas NAD^+ and NADH conserve the pyridine nucleotide. Conservation relations can be detected from the stoichiometry matrix by identifying linearly dependent rows [2, 24]. The number of independent conservation relationships can be determined directly by subtracting the rank of $\mathbf{N}$ from the number of internal metabolites in the system. Since the total concentration of a conserved moiety does not change with respect to time, Equation 1.8 can be used to describe the relationship between $\mathbf{N}$ and the set of conserved moieties:

$$\mathbf{N}^T.\tau = \mathbf{0} \tag{1.8}$$

where $\mathbf{N}^T$ is the transpose of $\mathbf{N}$ and τ is the conserved moieties matrix[14] and $\mathbf{0}$ is a zero column vector. Alternatively, conservation relationships can be calculated from the left null space of $\mathbf{N}$ [35]:

$$\mathbf{N}.\tau^T = 0^T \tag{1.9}$$

where τ^T is the left null space of $\mathbf{N}$ or the transpose of the conserved moieties matrix. For interpretive purposes, it is convenient that all coefficients in τ are non-negative in order to represent conservation of chemical units [36]. Sauro and

[13] from the model definition.

[14] rows represent metabolites participating in a conservation relationship and coefficients indicate a particular conserved sum with respect to the reactions in the network.

Ingalls described several methods and their corresponding algorithms for the determination of conservation relationships [26].

1.6.2 Null space analysis

Analytical procedures that fall into this category are those that involve the direct investigation of the kernel of the null space matrix.

1.6.2.1 Dead reactions

Dead reactions[15] also called strictly detailed balanced reactions [37] or blocked reactions [38] can be identified from the kernel of the null space as those reactions with all zero row entries. This indicates that they cannot carry flux for any possible steady state solution to the system. Further investigation of dead reactions could identify potential errors (i.e. incorrect reaction stoichiometries) or provide insights into the incompleteness of the system in question (i.e. missing reactions due to incorrect annotations) and thereafter, to maximise the number of reactions that produce flux by amending the model definition. This is an example of how a structural model can be iteratively refined (Chapter 4).

1.6.2.2 Enzyme subsets analysis

An enzyme subset is defined as a group of enzymes that carry flux in a fixed ratio at steady state [39, 40]. Despite this definition, an enzyme subset can actually be better described as a reaction subset since it is the reactions that carry flux through the subset, and additionally, an enzyme may catalyse more than one reaction, all of which may not be part of the same subset. For a linear system all reactions are part of a single subset, however, in more complex networks non-adjacent reactions may also belong to the same subset. By replacing those reactions in each subset with an overall reaction for the whole subset it is possible to reduce the size of the structural model. This not only aids in the interrogation of the model but also simplifies it for other more computationally intensive analysis methods (Section 1.6.3.1).

Proportional rows in the kernel of the null space are used to identify reactions within the same subset [1, 39]. Using Figure 1.4 as an example there are five subsets. These may be classified according to the number of reactions they contain i.e. subsets with a single reaction and those with two or more reactions. A kernel for this system is shown below:

[15] a term coined within our group.

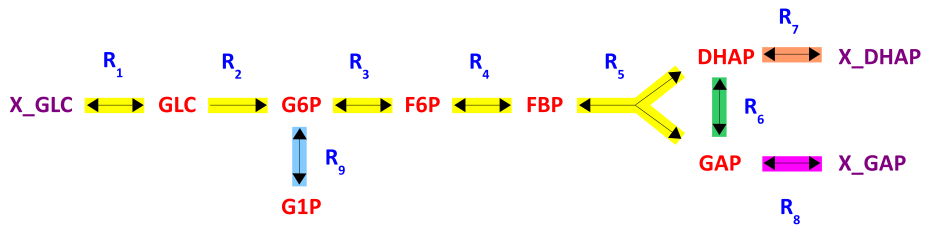


Figure 1.4 – Enzyme subsets as highlighted for an example reaction scheme derived from the top half of glycolysis leading to the production of glyceraldehyde-3-phosphate and dihydroxyacetone phosphate from glucose. Metabolites preceded by '*X_*' have been made external to the system. External ADP and ATP are included in reactions R_2 and R_4 but have not been included here for purposes of clarity. See List of Abbreviations section for metabolite abbreviations.

$$\mathbf{K} = \begin{array}{c} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \\ R_7 \\ R_8 \\ R_9 \end{array} \begin{pmatrix} -1 & 0 \\ -1 & 0 \\ -1 & 0 \\ -1 & 0 \\ -1 & 0 \\ -1 & 1 \\ -2 & 1 \\ 0 & -1 \\ 0 & 0 \end{pmatrix} \tag{1.10}$$

where each row represents a single reaction (R_1–R_9) and each column represents a possible solution space. It is evident that those reactions that are part of the same subset have proportional rows (R_1–R_5, R_6, R_7, R_8 and R_9). For the largest subset (R_1–R_5) the ratio of fluxes is 1:1 for any pair of reactions. Dead reactions show up in the same subset as those reactions with all zero row entries (R_9). The algorithm for detecting enzyme subsets as outlined in [39] is given below:

- Detect all row vectors of **K** that are null (i.e. imply 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. The quotient of the normalisation factors gives the flux ratio.

The ratio of fluxes obtained within a particular enzyme subset may indicate a proportional level of transcription at the genome-level. Using a genome-scale model of *E. coli*, Reed and Palsson [41] found a substantial correlation between the enzyme subsets from the model and the available gene expression data for the organism. Schuster *et al.* [42] showed that the enzyme subsets obtained for yeast central metabolism are correlated with expression data measured during the diauxic shift in the same organism. Moreover, within our group, enzyme subset analysis has been used with a number of microbial genomes to establish the relationship between the genetic regulation of a set of enzymes and their metabolic activities. Ultimately, these analyses may provide important clues as to the consistency of both the transcription and annotated genome data used in the subsets analysis [43].

1.6.3 Constraint-based analysis

The constraint-based analysis approach is based on the assumption that organisms exist in particular environments that typically have scarce resources [19, 27]. If the number of reaction fluxes is greater than the number of intracellular metabolites in a system, it is referred to as underdetermined and has infinitely many solutions. Therefore, additional constraints such as measurable fluxes can be used to reduce the number of unknown fluxes and, subsequently, to uniquely determine the flux distribution. In contrast to individual reaction rates, these measurements are largely available for metabolic networks. There are two fundamental types of constraints [19]:

- *Balances*
  Constraints that are associated with conserved quantities, such as energy, mass and redox potential.

- *Bounds*
  Constraints that limit numerical ranges of individual variables and parameters such as concentrations, fluxes or kinetic constraints.

By identifying and stating these constraints mathematically, they can be used to perform an *in silico* analysis. In mathematical terms, the range of allowable network states is contained within a solution space that represents the phenotypic potential of an organism [44, 45]. A set of valid solutions to Equation 1.5 subject to the defined constraints can be described within the solution space as a high-dimensional[16] polyhedral cone, by allowing the use of convex analysis [46] (Figure 1.5).

All structural analyses are constrained by the conservation of mass criteria imposed by the network stoichiometry. The analyses defined within this section

[16] number of dimensions equals the number of reactions.

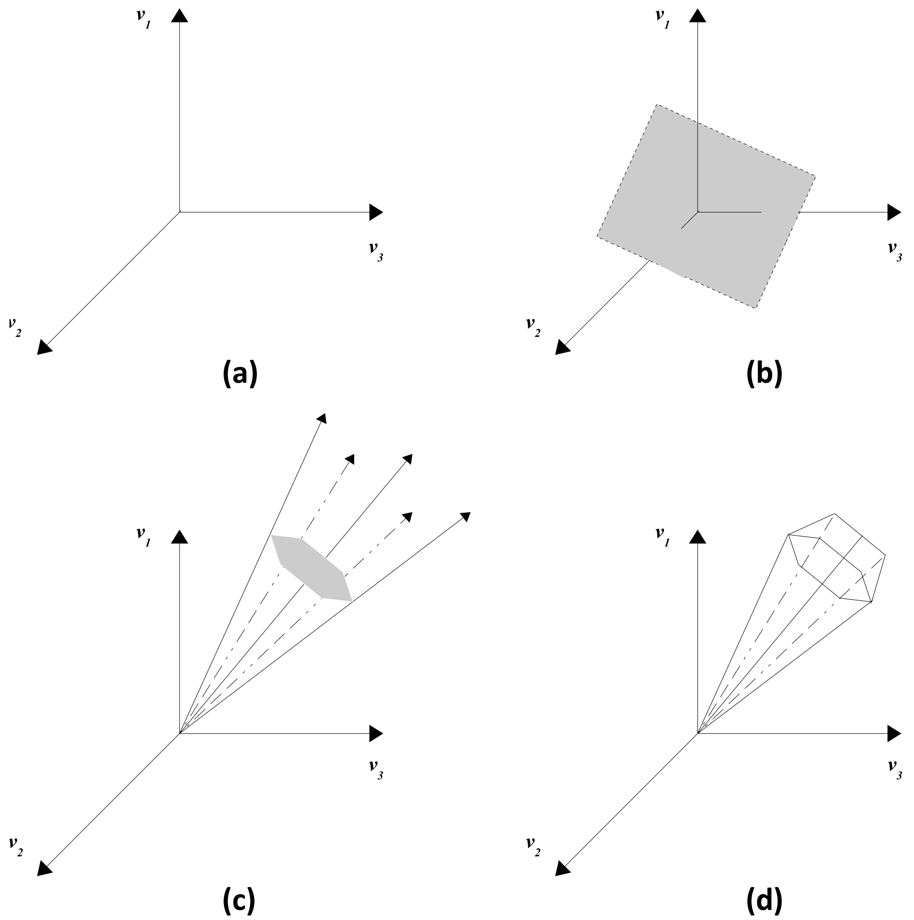


Figure 1.5 – By imposing successive constraints to a set of reactions in a metabolic network at steady state, it is possible to narrow down the range of possible solutions (i.e. flux distributions). (*a*) A three dimensional space where the axes represent individual fluxes through the network. (*b*) The mass balance constraint imposed by the stoichiometry matrix limits the steady state fluxes to a subspace. (*c*) If thermodynamic (i.e. irreversibility) constraints are taken into account so that all fluxes are positive this further limits the solution space to a convex cone. (*d*) Capacity constraints close the convex cone and can be searched for optimal solutions. Adapted from [47].

assume that under any given environmental condition, the system in question will reach a steady state that satisfies the following additional constraints in differing combinations:

1. *Reversibility* (*bound*)
   Only positive fluxes are allowed for irreversible reactions to satisfy thermodynamic criteria.

2. *Capacity* (*bound*)
   Defines lower and upper bounds for reaction capacities (e.g. maximum uptake rate of a transporter).

3. *Optimality* (*bound*)
   Minimise or maximise a linear objective function (e.g. optimise biomass

production).

By imposing constraint 1 on the network, it is possible to explore the phenotypic solution space for the feasible pathways (i.e. all possible steady state flux distributions) in the network [48]. Non-optimisation based techniques such as elementary modes analysis [49] and extreme pathways [50] fall into this category. With additional information on optimal network performance, metabolic flux analysis [51] and flux balance analysis [20] aim to identify alternate and usually more precise flux distributions according to some measured criteria. The scope of the work carried in this dissertation only employs elementary modes analysis, therefore, the remainder of techniques described above will be discussed to a supplementary extent.

1.6.3.1 Elementary modes analysis

Elementary modes (EMs) form a central theoretic concept for the structural analysis of metabolic pathways [23, 49, 52, 53]. An EM is a minimal set of reactions that could operate at steady state, such that all irreversible reactions in the mode are operating in the appropriate direction [52]. "Minimal" implies that if only the reactions belonging to this set were operating, removal of one of these would lead to cessation of any steady state flux in the rest of the mode. At steady state, an EM has no net consumption or production of any internal metabolite and, the reactions within the mode are weighted by the relative fluxes they need to carry for the mode to function. An EM may be thought of as a unique minimal route through a metabolic network which cannot be decomposed further to obtain other modes.

Convex analysis returns the spanning vectors of the convex solution cone that describes the steady-state equation system, and every actual flux distribution is a linear combination of the obtained EMs (Figure 1.5(*c*)). Several algorithms have been developed to calculate EMs [54, 55, 56]. The computation of EMs for larger genome-scale networks meets the problem of a combinatorial explosion in the requirement for computer memory and processing power. To demonstrate this issue Klamt *et al.* [40] built a model containing 110 reactions and 89 metabolites involved in *E. coli* central metabolism. With the use of four external source metabolites (i.e. glucose, acetate, succinate and glycerol) the model was found to have 507,632 EMs. Consequently, a number of preprocessing steps have been suggested to improve the efficiency of the calculation procedure including:

- lumping reactions into their enzyme subsets before the calculation reduces the size of the network [54]. This has no effect on the resulting pathways since the individual reaction information can be restored after the calculation.

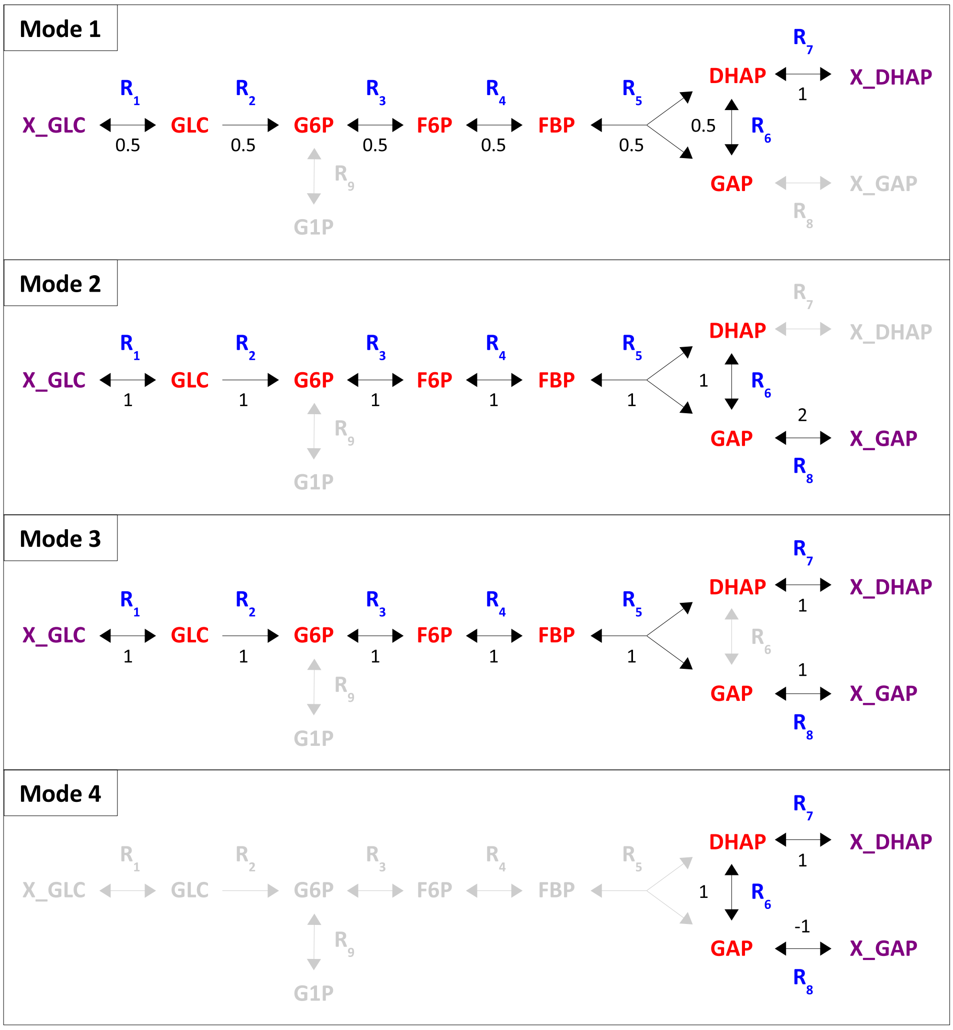

Mode 1
R1 R2 R3 R4 R5 R6 R7 R8 R9
X_GLC GLC G6P F6P FBP DHAP GAP X_DHAP X_GAP G1P
0.5 0.5 0.5 0.5 0.5 0.5 1
Mode 2
R1 R2 R3 R4 R5 R6 R7 R8 R9
X_GLC GLC G6P F6P FBP DHAP GAP X_DHAP X_GAP G1P
1 1 1 1 1 1 2
Mode 3
R1 R2 R3 R4 R5 R6 R7 R8 R9
X_GLC GLC G6P F6P FBP DHAP GAP X_DHAP X_GAP G1P
1 1 1 1 1 1 1
Mode 4
R1 R2 R3 R4 R5 R6 R7 R8 R9
X_GLC GLC G6P F6P FBP DHAP GAP X_DHAP X_GAP G1P
1 1 -1

Figure 1.6 – The four possible EMs for the simple metabolic network introduced in Figure 1.4. Flux values are indicated on the reactions that participate in each mode.

- metabolites with a connectivity above a certain user-defined threshold can be made external to split the network into sub-networks, which are easier and possibly more convenient to analyse [5].

The various computational issues for generating the modes are compounded with the problem of analysing and assigning biological significance to the vast amount of data generated. Using Figure 1.4 as an example system there are four possible EMs (Figure 1.6). Excluding net ATP to ADP conversions the first mode uses half a glucose (GLC) to produce one dihydroxyacetone (DHAP) (flux of half for R_1–R_6 and one for R_7). The second is the same except that all stoichiometries are doubled and glyceraldehyde-3-phosphate (GAP) is produced instead of DHAP (flux of one for R_1–R_5, minus one for R_6 and two for R_8). The third produces one each of DHAP and GAP by utilising 1 GLC (flux of one for R_1–R_5 and R_7–R_8). The fourth mode is the simplest, and utilises external GAP to produce external DHAP (flux of one for R_6–R_7 and minus one for R_8). Additionally, all modes except for the fourth are irreversible since they are involved with at least one irreversible reaction (R_2). The carbon balance is as would be expected *in vivo* for a six carbon sugar (i.e. GLC) going to three carbon sugars (i.e. GAP and DHAP). From this example, it has become evident that an EM is simply a route for the conversion of external metabolites. There is sometimes an exception to this generalisation whereby a mode may be formed by complete balancing of only internal metabolites. In other words, by starting and ending at the same internal metabolite via two or more reactions. Such modes are called futile (or substrate) cycles since they are thought to be biologically wasteful [57].

Elementary modes analysis (EMA) has been used in many model systems to interrogate various aspects of network topology. The number of EMs may be an important index in order to characterise the functional richness and capabilities of biochemical systems. Therefore, it is also possible to determine the level of structural versatility in a given network using EMA. Stelling *et al.* [58] demonstrated this concept from a model of *E. coli* central metabolism. Poolman *et al.* [59] built a model of chloroplast metabolism (i.e. Calvin cycle and oxidative pentose phosphate pathway enzymes) to compare light/dark metabolism using EMs.

EMA has proved to be very useful for the computation of maximal conversion yields by investigating all the possible routes from a particular substrate to product. Patnaik *et al.* [60] used EMA in *E. coli* to identify ways in which to maximise the yield of aromatic amino acids by increasing the availability of central metabolites such as phosphoenolpyruvate. Carlson *et al.* [61] used a recombinant *E. coli* system to study the effect of altered culturing conditions to optimise poly-(R)-3-hydroxybutyric acid yield using EMA.

1.6.3.2 Extreme pathway analysis

Extreme pathways (EPs) are a unique and minimal set of vectors that completely characterise the steady-state capabilities of any given metabolic network [50]. Extreme pathways analysis (EPA) and EMA both return the edges of the convex solution cone for the network as pathways (Figure 1.5(c)). In addition, EMA also returns all the possible non-decomposable pathways through the network. Extreme pathways and EMs share the following properties:

- There is a unique set for a given network.
- They are non-decomposable, therefore, if a reaction is removed from an EP or EM then the steady-state flux through it becomes zero.

EPs have an additional property of systemic independence, which means that an EP cannot be represented as a non-negative linear combination of any other EP.

When using EPA, negative reaction fluxes are avoided by splitting all reversible internal reactions into two separate irreversible reactions. EPs form a subset of the EMs (i.e. combination of extreme pathways), with the exception that if all the reactions in the network are irreversible then identical sets are obtained [62]. Therefore, the number of EPs is less than or equal to the number of EMs, however, the computational issues realised with EMA also applies to EPA [50, 53]. The subtle differences between the theoretical definitions of EPs and EMs can lead to different descriptions of network properties, such as pathway redundancy (due to differing number of EP and EM sets) and the yield of products from substrates. Several recent publications discuss the difference between EPA and EMA [48, 62, 63]. EPA has been used to show that there is more redundancy in the production of amino acids by the *Haemophilus influenzae* metabolic network than in the *Helicobacter pylori* metabolic network [64]. The complete set of EPs has been obtained for human red blood cell metabolism and has been used to conservatively predict values that describe the maximal ATP:NADPH yield ratios the cell can sustain under a load, calculated from the kinetic model [22].

1.6.3.3 Metabolic flux analysis

At steady state, metabolic flux analysis (MFA) aims to shrink the possible solution space of Equation 1.5 by inclusion of measured reaction rates (i.e. uptake or excretion rates) [51]. The measured reaction rates are then used to calculate a subset of all the unknown rates. For the purposes of MFA, Equation 1.5 becomes:

$$0 = \mathbf{N}\mathbf{v} = \mathbf{N}_b\mathbf{v}_b + \mathbf{N}_n\mathbf{v}_n \tag{1.11}$$

$$\mathbf{N}_n\mathbf{v}_n = -\mathbf{N}_b\mathbf{v}_b \tag{1.12}$$

where $\mathbf{N}$ and $\mathbf{v}$ are partitioned into known ($\mathbf{N}_b\mathbf{v}_b$) and unknown ($\mathbf{N}_n\mathbf{v}_n$) parts. All unknown rates ($\mathbf{v}_n$) can only be calculated if the stoichiometry matrix $\mathbf{N}_n$ is a square matrix (number of unknown reactions is equal to the number of unknown metabolites) and invertible. In reality, all unknown rates are rarely calculable, however, in some cases it is possible to determine the values of a subset of the unknown rates [65, 66]. Klamt *et al.* [66] developed an improvement on the algorithm by van der Heijden *et al.* [65], with which to find the reaction rates that can be uniquely calculated in underdetermined metabolic networks. In the same study, Klamt *et al.* applied their algorithm to a metabolic model of the central metabolism in purple nonsulfur bacteria. ^{13}C labelling has also been used to determine the intracellular fluxes of *Corynebacterium glutamicum* for use in MFA to calculate all reaction rates [67].

1.6.3.4 Flux balance analysis

Additional constraints are imperative to uniquely determine the steady-state flux distribution for underdetermined systems. Like all constraint-based approaches, flux balance analysis (FBA) [20] aims to further restrict the solution space imposed by the mass balance constraint from the stoichiometry of the system. Commonly, ranges of allowable flux values are incorporated as additional constraints to resemble the performance capability of the network. Subsequently, the system is assumed to be optimised in order to find the optimal value of a specified objective function[17] (i.e. maximisation of biomass production, minimisation of ATP utilisation or any other estimated function) [68]. Linear programming (LP) is the method of choice to calculate the maximum potential of the objective function, and therefore, when using FBA, a single solution or flux distribution is found [69]. Furthermore, the metabolic capabilities of the system can then be investigated using this flux distribution.

The outcomes from FBA are highly dependent on the declared constraints and, while specifying known flux data reduces the possible flux space, it still allows for infeasible phenotypic predictions. In addition, LP does not guarantee a unique solution and, numerous solutions can be found which satisfy the metabolic network constraints and optimal objective value [19]. Methods for interpreting alternate optima have been developed and include:

- *Mixed-Integer Linear Programming* (MILP)
  Used to find all the extreme points of the convex polytope (the feasible constrained solution space) that have the same optimum objective function value [70]. The method has been used to design NMR experiments for measuring *in vivo* intracellular fluxes for the central metabolic network of *E. coli* [71].

- *Minimisation of Metabolic Adjustment* (MOMA)
  Employs a quadratic programming approach for predicting metabolic flux distributions after gene knockout [53]. Subject to the new constraints imposed, the algorithm aims to finds a point (i.e. flux distribution) in the altered solution space that is closest to an optimal point in the wild-type solution space [68].

- *Flux Variability Analysis*
  Determines the maximum and minimum values (i.e. feasible range) of each flux in the network, while still satisfying the given constraints and optimising a particular objective function [72]. Using this approach the range of flux

[17] represents probable physiological function, and is defined in context of the system under investigation.

variability is identified as opposed to all alternate optima and can be used to study the entire range of achievable cellular phenotypes [72].

The metabolic capabilities of several organism including *Saccharomyces cerevisiae* [73] and *H. influenzae* [74] have been studied using FBA. For example, network robustness can be explored by varying the maximum flux through a particular reaction and observing the resultant growth rate [69]. Edwards and Palsson [75] used this approach to conclude that *E. coli* is robust to changes in individual enzyme or pathway activities. Gene knockout (i.e. knockout reaction flux is constrained to zero) and addition studies can be used to study the resultant phenotypic consequences by reducing and expanding the wild-type solution space, respectively [69]. Gene knockouts have been evaluated extensively in *E. coli* [20, 68] and *Staphylococcus aureus* [76].

1.7 Structural modelling: software

The use of programming resources in stoichiometric analysis is an absolute necessity. It is simply not possible, except in the most trivial cases, to carry out the computations by hand. Moreover, with the use of the large datasets that have been made available through bioinformatics techniques, a genome-scale metabolic model may typically contain in excess of 250 reactions. When developing a bioinformatics analysis pipeline, it is more important to have a good understanding of both the biology involved and the analytical techniques rather than having the right software.

Various software packages have been developed for the purposes of kinetic modelling including Copasi[†] [77] (formerly Gepasi[†] [78]) and Jarnac[†] [79] (upgraded version of SCAMP[†] [80]). The Python Simulator for Cellular Systems (PySCeS[†]) [81] is similar to our in-house software, ScrumPy, in that it is written in the Python programming language[†] [82]. The way in which PySCes has been developed allows the user to dictate the flexibility and extendibility of his/her own research, another feature which is common to ScrumPy. PySCes is an open source console-based application that can be installed on both Windows and Linux operating systems. The majority of modelling functionality implemented in PySCes makes use of existing SciPy[†] libraries, which are a large collection of mathematical algorithms for science and engineering applications. Extensive functionality for kinetic modelling is provided by PySCes with some additional but little support for structural modelling (uses Metatool for EMA).

Dynamic simulation packages have very few options for structural analytical techniques and, therefore, several packages can be consulted for these purposes.

Metatool† [39] was one of the first programs developed for sole dedication to the stoichiometric analysis of metabolic networks. It is a command-line driven program written in the C† programming language and until the development of its successor YANA† [83] the results were outputted as a text file containing information for elementary modes, enzyme subsets and conserved moieties. YANA is a platform-independent graphical user interface (GUI) written in Java† for metabolic network analysis, to calculate (integrating Metatool), edit (SBML† support provided), visualise, and compare EMs.

CellNetAnalyzer† [84] is a GUI that integrates the mathematical functionality of the commercially available package MATLAB† by providing an extensive toolbox for the structural and functional analysis of cellular networks. In contrast to its predecessor, FluxAnalyzer [85], CellNetAnalyzer can be used to analyse signalling and regulatory networks as well as metabolic networks. Most stoichiometric analyses can be carried out including EMA (calculated using Metatool) and FBA.

1.7.1 Metabolic modelling with Python: ScrumPy

ScrumPy† [86] is an open source, metabolic modelling software package that has been implemented in our lab using the Python programming language. ScrumPy supports both kinetic and structural modelling and is currently available for Linux and other Unix-like systems.

1.7.1.1 The Python programming language

The Python [82] programming language has been developed recently when compared to other languages such as C or Java. It is an open source, object-orientated and platform independent language that is easy to learn, due to its simple syntax and console-based interactive development environment. Python has simple but efficient tools for handling its powerful built-in type 'objects' (Table 1.7.1.1). Programmers can define their own objects in the form of 'classes', which are typical for object-orientated programming. Object functionality is created by defining 'methods' within the class structure and once the class has been instantiated (e.g. `instance()`) methods can be called using 'dot' notation (e.g. `instance.method(argument)`; where `argument` is a variable or value passed into `method`).

When using other programming languages such as Java, a programmer would have to explicitly specify memory-recycling events for objects. In contrast, Python automatically allocates and reclaims objects in memory when they are no longer in use. Furthermore, designing and writing the same program took half as much time as writing it in C, C++ or Java, and the resulting program was half as long

Type	Description	Syntax Example
`str`	An immutable[a] sequence of characters	`‘This is a string’`
`int`	Integer	`42`
`float`	Floating point	`3.141592`
`bool`	Boolean	`True` or `False` (`1` or `0`)
`tuple`	Immutable, can contain mixed types	`(‘string’, 7.7, True)`
`list`	Mutable, can contain mixed types	`[‘string’, 7.7, True]`
`dict`	Group of key and value pairs	`{‘key1’:5, ‘key2’:[‘no’, 7.7]}`

[a] an object whose state cannot be modified after it is created.

Table 1.1 – Important Python built-in types and examples.

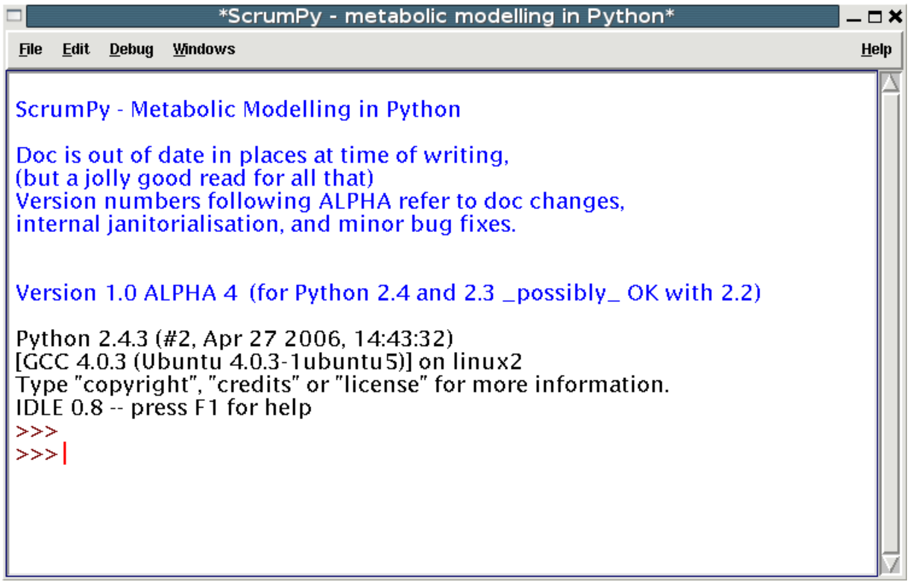
[87]. Although, languages such as C and C++ offer a more efficient platform for tasks that need to handle large amounts of computation and data, Python offers significant advantages with respect to programmer productivity. However, it is possible to integrate C and C++ code with Python using software development tools such as SWIG[†].

Other than those in the standard Python distribution, a diverse library of supplementary packages are also available for mathematical and scientific programming (e.g. NumPy and SciPy). Additionally, a Python-based package called BioPython[†] has been developed which can be used to automate tasks such as sequence analysis and biological database parsing.

1.7.1.2 Why ScrumPy?

ScrumPy has been developed to provide a high-level metabolic modelling interface that exploits and extends the low-level capabilities provided by Python. As with PySCes, the functionality provided by ScrumPy can easily be extended by writing Python programs for custom application to the modelling process. In contrast, stand-alone GUI-based packages such as Gepasi limit the user to the functionality provided by the creators of the software.

Along with a GUI component for kinetic modelling, ScrumPy also enables the user to benefit from the interactive command-line component provided by Python for all modelling and development purposes. Once ScrumPy has been started a window such as the one in Figure 1.7 is opened. This represents the front-end to ScrumPy and is actually an adapted version of the Python Integrated Development Environment interface[†] (IDLE). Thus, customisation of the IDLE interface has permitted the simultaneous use of an existing Python development GUI with the functionality add-ons of ScrumPy's metabolic modelling package. The reader is directed to Appendix A for a basic guide to the structural modelling capabilities of ScrumPy.



```
ScrumPy - Metabolic Modelling in Python

Doc is out of date in places at time of writing,
(but a jolly good read for all that)
Version numbers following ALPHA refer to doc changes,
internal janitorialisation, and minor bug fixes.


Version 1.0 ALPHA 4  (for Python 2.4 and 2.3 _possibly_ OK with 2.2)

Python 2.4.3 (#2, Apr 27 2006, 14:43:32)
[GCC 4.0.3 (Ubuntu 4.0.3-1ubuntu5)] on linux2
Type "copyright", "credits" or "license" for more information.
IDLE 0.8 -- press F1 for help
>>>
>>>
```

Figure 1.7 – An IDLE shell representing the ScrumPy development environment. The blue text represents ScrumPy version information, the black text is a Python welcome message and the red '>>>' is a prompt awaiting a Python command from the user.

The ability to use high-level modelling modules as well as low-level Python-based functionality is a great advantage to any modeller, and requires minimal learning effort. The flexible nature of ScrumPy will facilitate the development of novel ways of metabolic modelling and, ultimately, in understanding complex cellular behaviour albeit structural or kinetic.

1.8 Overall aims and objectives

The core of this project involved the use of metabolic modelling techniques to investigate the network properties for models of varying size. Chapter 2 introduces hierarchical clustering techniques for application to the interpretation of the large datasets that may arise from structural analytical procedures, more specifically, elementary modes analysis (Section 1.6.3.1). From the clustering output, alternative visualisation methods will be investigated alongside the more traditional dendrograms, and will initially be validated on a small model of yeast metabolism. Thereafter, in Chapter 3 these techniques will be applied to a more extensive model encompassing the flow of carbon through the central metabolism of *S. erythraea*. The primary objective of these studies will be to assess the application of hierarchical clustering for the grouping and, subsequent biological interpretation of elementary modes.

As highlighted in Section 1.3, the increased availability of both conventional biochemical and genome annotation data has been exploited as a platform for the reconstruction of genome-scale models of metabolism. Consequently, the acquisition of reaction data from pathway databases, and its subsequent translation into a format suitable for modelling has become technically trivial. However, the major difficulty lies in the quality of the resulting metabolic network and its limited ability to reflect the properties of the real organism at the systems-level. Chapter 4 will report a group effort to address the problems encountered therein, and to identify potential steps that can be taken to improve the quality of the model in order to generate more reliable phenotypic predictions.

In addition to traditional 16S ribosomal RNA-based phylogenies, a number of techniques have been developed to exploit data on a genome-scale to build evolutionary relationships amongst organisms. In Chapter 5, phylogenetic trees will be generated from species-specific enzyme complement using the customised programming tools already developed for hierarchical clustering and automated genome-scale model reconstruction. A comparison of the resulting phylogenetic trees with 16S rRNA-based trees will be carried out to highlight interesting phenotypic and taxonomic discrepancies. More notably, this study will be carried out on the largest prokaryotic dataset used thus far and may, therefore, help to clarify the results from other studies, and the opportunity to determine the degree of metabolic similarity between various fully sequenced prokaryotic species. As might be expected, the enzyme complement trees are limited in their ability to generate phylogenetic predictions by the amount and quality of the data included for individual species. In summary, the investigations reported herein will be used to highlight how co-ordinated research efforts at the systems-level may need to be reevaluated in order to increase the potential knowledge that can be gained from building and interpreting genome-scale models of metabolism.

CHAPTER 2

Interpreting Elementary Modes Using Cluster Analysis

2.1 Introduction

Elementary modes analysis (EMA) is a very useful technique to assess the structural and functional capabilities of metabolic networks (Section 1.6.3.1). The potential for a large elementary modes (EMs) dataset even for small models (20–30 reactions) somewhat undermines the usefulness of EMA. Once generated, there are currently no well known methods in place to aid in the understanding and grouping of EMs to determine their biological significance. Of particular interest for this project is the use and validation of hierarchical clustering for the visualisation and interpretation of EMs. Hierarchical clustering techniques have been used extensively for the visualisation and analysis of gene expression data by identifying genes with similar expression patterns that are assumed to have a functional relationship [88]. Similarly, clustering techniques may be useful to uncover groups of EMs that are functionally related (i.e. in terms of their reaction profiles or net stoichiometry) at the metabolic-level. Additionally, as opposed to looking at an unordered or ungrouped set of EMs, the interpretation of larger EM datasets may be improved by investigating smaller subsets of related EMs that form part of the whole set.

2.2 Data mining

The process of extraction of previously unknown, meaningful information from large datasets is known as data mining [89]. Machine learning provides the technical basis of data mining and deals with the design and development of automated computational and statistical methods for data interrogation [90]. Machine learning can be further divided into two primary sub-fields:

- **Supervised**
  Algorithms of this type are exposed to a functionally related training set of data points (i.e. known inputs and outputs) and their respective classification

categories (i.e. labels). The goal is then to predict the label of any new valid input object. Supervised learning techniques cannot be applied to this study, primarily because the classification information required for the data is unknown. Techniques include neural networks and, more recently, support vector machines [91].

- **Unsupervised**
  As suggested by its name, and in contrast to supervised learning, the dataset in question is not provided with any supplementary class information. Instead the main aim of this family of algorithms is to uncover the inherent structure of classes within the dataset. The most commonly employed unsupervised classification methods are the clustering techniques, which is the focus of this chapter.

2.3 Cluster analysis

The term cluster analysis, also called taxonomy analysis, encompasses the classification of a dataset into groups (clusters) based on some measure of similarity between individual items. A cluster is therefore, a collection of data items which are more alike when compared to those in other clusters. The extent of applicability of a given clustering method requires the ability to define meaningful difference values between items in the dataset, and to deal with large datasets, different types of attributes, outliers and high data dimensionality. Additionally, it should be easy to implement, use and interpret.

2.3.1 Clustering algorithms

The plethora of methods for the representation, distance measurement and grouping of individual dataset items has led to a vast collection of clustering algorithms in the literature [92]. As a consequence, a user attempting to find an algorithm suitable within a given domain of expertise is often overwhelmed. Domain information regarding the data at hand, knowledge of the broad categories of clustering techniques, as well as the required clustering outcomes are very important for assessing the true class structure of the dataset [93].

The most popular clustering algorithms fall into two classes, namely hierarchical and partitional. The results from hierarchical clustering algorithms can be visually interpreted in the form of a dendrogram, which significantly contributes to their popularity of use (Figure 2.1). A dendrogram is a simple and compact representation of the dataset, formed by a series of nested partitions, with individual elements at one end and a single cluster containing every element at the

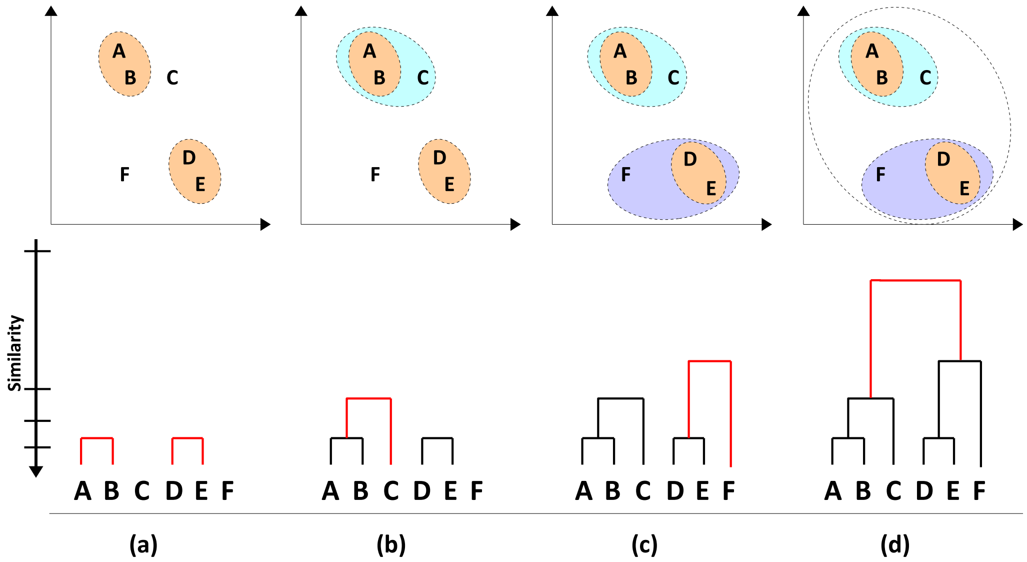


Figure 2.1 – Diagrammatic representation of a generalised hierarchical clustering algorithm for six points (A–E) in two dimensions. Clusters are shown to be formed in the space occupied by the points (above) and in parallel as a dendrogram (below). Adapted from [94].

other (i.e root of the tree). The closer a pair of attributes are within the tree the more similar they are and vice versa. Figure 2.1 can be used to demonstrate a generalised hierarchical clustering algorithm on 6 points in two dimensional space:

1. All points begin in a cluster of their own. Merge the closest pairs of data points successively (A–B and D–E) which is also reflected in the dendrogram as a link with height reflecting the similarity between data points (Figure 2.1(a)).

2. The next closest clusters (A–B and C) are merged to form a new cluster (Figure 2.1(b)).

3. The former step is repeated until only one cluster remains (Figure 2.1(c)-(d)).

Breaking the hierarchy at desired levels can be used to obtain different clusterings of the data, where each connected component forms a cluster. This type of differentiated output is typical of the most popular partitional algorithm, k-means analysis [95] whereby the algorithm partitions all the data items into a user defined number of clusters with each data item belonging only to one cluster. Using Figure 2.2 as an example to demonstrate the k-means algorithm on 6 points in two dimensional space:

1. Define the number of required clusters, k (i.e. in this case, k=2) and randomly place k points into the space represented by the data points being clustered, which represents the initial group centres (crosses) (Figure 2.2(a)).

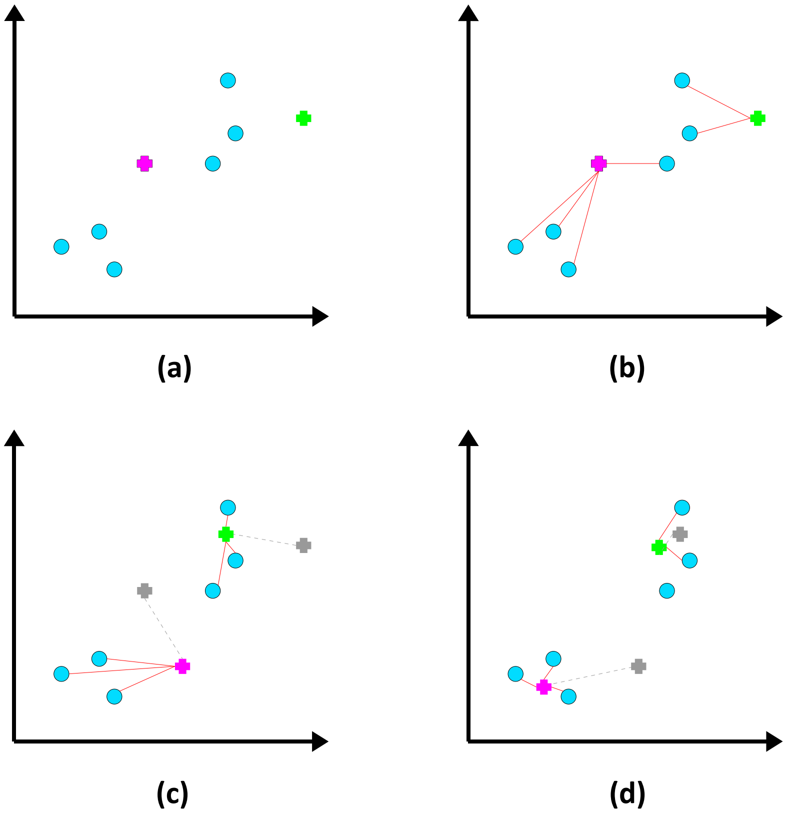


Figure 2.2 – Diagrammatic representation of a generalised k-means algorithms as demonstrated on 6 points (circles) in two dimensions. Adapted from [94].

2. Assign each data point to the closest centre (red lines) (Figure 2.2(b)).

3. Centres are moved to the average position of all the points in its cluster and repeat the former step for any change in assignment (Figure 2.2(c)).

4. The termination condition is reached since the centres do not move (Figure 2.2(d)).

Therefore, as opposed to a dendrographic visualisation obtained from hierarchical clustering, the final output from k-means analysis is a list of k clusters and their corresponding data points without any indication as to the distance between individual clusters or the data points within them.

2.3.2 Hierarchical vs. k-means clustering

The main advantage of the partitional k-means algorithm is its speed, realised by assigning data points to a predefined number of clusters (Figure 2.2), whereas hierarchical methods find successive clusters using previously established ones (Figure 2.1). The k-means algorithm is also very easy to implement, however, it has a number of drawbacks:

- The number of output clusters (k) have to be defined before running the algorithm. This poses problems since it is difficult to predict the optimal value of k in advance for any given dataset. Possible solutions to this problem include running the algorithm multiple times with varying values of k. The best value of k will provide the largest inter- and smallest intra-cluster distances, respectively. However, the converse of this argument suggests that a defined number of clusters may provide optimal separation of the dataset for further interrogation.
- It is very sensitive to the positions that each of the random k initial partitions are placed [96] (Figure 2.2(a)). Ideally, the initial centres should be placed close to the centre of the natural clusters. However, this is very difficult without prior knowledge about the clusters, which means that the algorithm often converges to a local minimum. In contrast, hierarchical clustering always returns the same unique result.
- Multiple iterations with different initialisations are usually carried out to overcome the former problem. The most frequent result is usually chosen and is likely to indicate that a global minimum has been reached. Depending on the number of iterations chosen and possibly the size of the dataset, the speed advantage obtained by the k-means algorithm may be nullified by a single run of the hierarchical algorithm.

Limitations of the hierarchical approach are usually manifested when dealing with larger datasets (i.e. increased computation times), even so, when compared to k-means [97] they produce a much better quality of clustering by always returning a unique dendrogram. In the context of this dissertation, considering the size of the datasets to be used, the critique above shows that there is no distinct advantage for employing k-means, and where necessary all clustering investigations carried out herein will use hierarchical algorithms.

2.4 Hierarchical clustering

There are two types of hierarchical clustering techniques depending on whether the clustering process is initiated at the root or the tips of the tree. Agglomerative algorithms start at the top of the tree with each data point in a cluster of its own and iteratively merge them to obtain a hierarchy (Figure 2.1). Divisive algorithms do the opposite by beginning with all data points in one cluster and then breaking it down into smaller subsets until each subset consists of a single data point. Divisive methods are rarely used since they are more computationally expensive and will not be considered further.

$$\begin{array}{c} A \\ B \end{array} \begin{pmatrix} 0 & 3 & 4 & 5 \\ 7 & 6 & 3 & -1 \end{pmatrix}$$

Figure 2.3 – An example dataset to illustrate the calculation of distance measures. Each row represents the data point to be clustered and each column represents their individual features. Each element can be represented as a point in 4 dimensions where A and B have coordinates (0, 3, 4, 5) and (7, 6, 3, -1), respectively.

2.4.1 Distance measures

The definition of a cluster is based on the similarity between data points, therefore, clustering algorithms require the selection of a suitable distance measure, which will determine the pairwise similarity for all the chosen data points. A separate distance matrix (i.e. $n \times n$) can be calculated from the original data matrix containing the raw data values to be clustered (e.g. 2×2 matrix for data in Figure 2.3). Depending on the distance measure employed, the entries in the matrix are a measure of similarity for any pair of data points, with all zero values on the leading diagonal indicating that the distance between the same data point is zero.

Different distance measures will produce different clustering results, therefore, finding a good distance measure depends on the dataset in question. Euclidean distance is the most commonly used distance measure, although as highlighted in the next section, several other methods have found application in various communities [97].

2.4.1.1 Absolute distance measures

Measures of this type are involved in the calculation of the actual projected distance between any two data points in n-dimensional space. The Minkowski distance is a generalised distance function used to derive other more common measures. It defines the distance between any two points as:

$$d_{ij} = \left(\sum_{k=1}^{n} |x_{ik} - x_{jk}|^{p} \right)^{1/p} \tag{2.1}$$

where n is the number of dimensions and the value of p indicates the type of distance in the formula. Differing values of p give different variations of the Minkowski distance:

- *Manhattan or City-block distance* ($p = 1$)
  Measures the absolute difference between coordinates of a pair of objects (Figure 2.4). Equation 2.1 becomes:

$$d_{ij} = \sum_{k=1}^{n} |x_{ik} - x_{jk}| \tag{2.2}$$

Using the example dataset in Figure 2.3:

$$\begin{aligned} d_{BA} &= |0-7| + |3-6| + |4-3| + |5+1| \\ &= 7+3+1+6 \\ &= 17 \end{aligned}$$

- *Euclidean distance* ($p = 2$)
  The most commonly used distance measure, calculates the root of square differences between coordinates of a pair of objects (Figure 2.4). Equation 2.1 becomes:

  $$d_{ij} = \left(\sum_{k=1}^{n} |x_{ik} - x_{jk}|^2 \right)^{1/2} \tag{2.3}$$

  Using the example dataset in Figure 2.3:

  $$\begin{aligned} d_{BA} &= \sqrt{(0-7)^2 + (3-6)^2 + (4-3)^2 + (5+1)^2} \\ &= \sqrt{49+9+1+36} \\ &= 9.747 \end{aligned}$$

- *Chebyshev distance* ($p = \infty$)
  Also called maximum value distance, determines the absolute magnitude of the differences between coordinates of pairs of objects and returns the maximum value (Figure 2.4). Equation 2.1 becomes:

  $$d_{ij} = \max_{k} |x_{ik} - x_{jk}| \tag{2.4}$$

  Using the example dataset in Figure 2.3:

  $$\begin{aligned} d_{BA} &= \max\{|0-7|, |3-6|, |4-3|, |5+1|\} \\ &= \max\{7, 3, 1, 6\} \\ &= 7 \end{aligned}$$

Minkowski-derived metrics tend to work well with datasets that have compact or isolated clusters, however, they tend to give excessive weight to objects further apart (i.e. outliers) [98]. The selection of a suitable measure for a given application is by no means standardised and depends on the discretion of the user or its availability in an analysis software of choice. As indicated later, ScrumPy provides an option to generate a distance matrix using arbitrary Python functions that represent distance measures of choice.

2.4.1.2 Correlational measures

Correlational measures are used to calculate the similarity as opposed to the absolute distance or dissimilarity between any two data vectors. Furthermore, correlational measures do not place any emphasis on the size of the vectors being compared, instead, they capture the similarity in shape of their overall profiles (Figure 2.5). The most frequently used measure in this category is the Pearson's correlation coefficient. When comparing normalised vectors the uncentred Pearson's correlation coefficient is identical to the cosine of the angle ($\cos\theta$) between two vectors and lies within the range 1 to -1. Furthermore, the angle (θ) between the vectors (Figure 2.4) can be determined from $\cos\theta$ to indicate the similarity between the vectors in n-dimensions. θ can be calculated from the cos inverse of the dot product of the two vectors divided by the product of the length of each vector:

$$\theta = \cos^{-1}(\frac{a \cdot b}{\|a\|\|b\|}) \tag{2.5}$$

Using the example dataset in Figure 2.3:

$$\begin{aligned}
\cos(A,B) &= \frac{0\cdot 7 + 3\cdot 6 + 4\cdot 3 + 5\cdot(-1)}{\sqrt{0^2+3^2+4^2+5^2}\sqrt{7^2+6^2+3^2+(-1)^2}} \\
&= \frac{0+18+12-5}{\sqrt{0+9+16+25}\sqrt{49+36+9+1}} \\
&= \frac{25}{\sqrt{50}\sqrt{95}} \\
&= 0.363 \\
\theta &= cos^{-1}(0.363) \\
&= 1.2\ radians
\end{aligned}$$

The relationship between θ and $\cos\theta$ can be interpreted from the graph of the cosine function whereby identical vectors have $\theta = 0$ and $\cos\theta = 1$, orthogonal vectors (i.e. $\theta = \frac{\pi}{2}$) have $\cos\theta = 0$, and vectors in the opposite directions will have $\theta = \pi$ and $\cos\theta =$ -1.

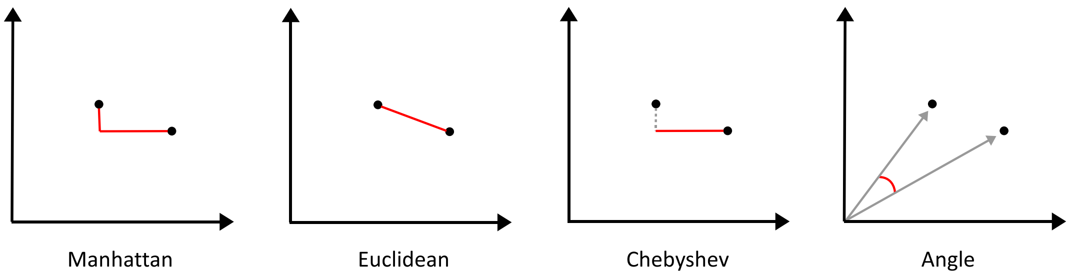


Figure 2.4 – Illustration to show how various distance measures can be represented diagrammatically for two points in two dimensions. Adapted from [94].

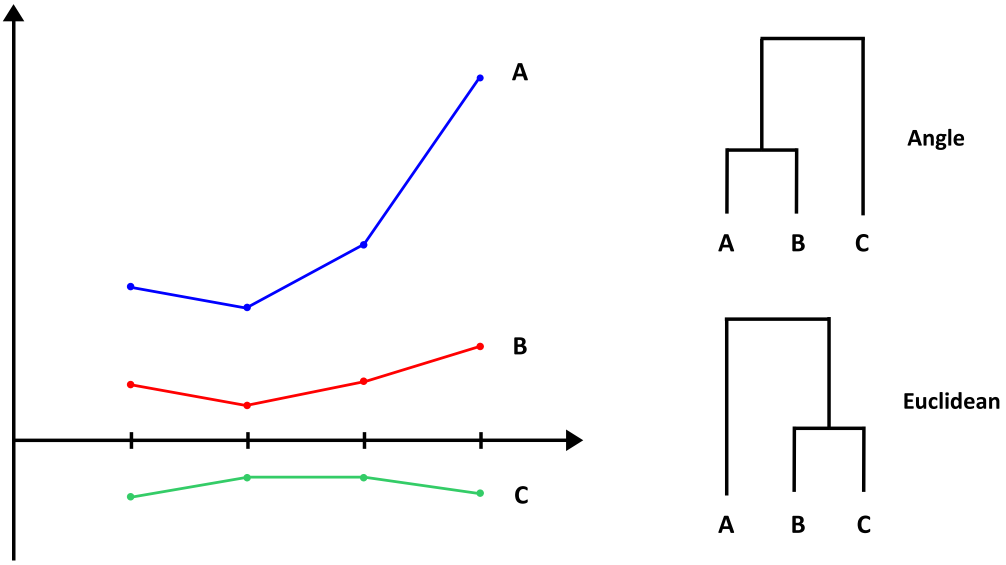


Figure 2.5 – Illustration to show the difference between absolute and correlational distance measures. As reflected in the dendrogram (right) the three data vectors (A–C) will be clustered in different ways. The euclidean distance will group B and C together based on their lower absolute difference, whereas clustering by angle will group A and B due to the similarity in their overall profiles [99].

2.4.2 Agglomerative algorithms

Using a measure of choice, once a distance matrix has been generated for individual data points there are several variants of agglomerative algorithms depending on the way in which the distance between individual clusters is defined. Three of the most popular methods include [97]:

- *Single-linkage*

  Defines the difference between two clusters as the minimum distance between any member of one cluster to any member of the other cluster (Figure 2.6(a)). This method suffers from producing large, elongated clusters (i.e. the chaining effect), since it forces data points to be close to each other without considering all the other points in the cluster. However, the same behaviour is useful for detecting outliers in the dataset.

- *Complete-linkage*
  The opposite of single-linkage, this algorithm defines the difference between two clusters as the maximum distance between all pairs of data points drawn from them (Figure 2.6(b)). As expected, the clusters obtained using this method have the opposite characteristics when compared to the single-linkage technique (i.e. smaller and more compact). Complete-linkage algorithms should not be applied to noisy datasets since they are more sensitive to outliers.

- *Average-linkage*
  Also called unweighted pair group method using arithmetic averages (UPGMA) [100], variants of this method tend to be a compromise between the single- and complete-linkage methods. Similarity is determined by finding the average value of all the data points in each cluster, and grouping those that have the smallest average distance between them (Figure 2.6(c)). It is more computationally expensive, however, it is the most popular linkage method, since it provides a trade-off in terms of the symptoms that are characteristic of single- and complete- linkage.

When compared on the same dataset, side-by-side, each of the methods described above are very likely to produce different results. Therefore, an individual method cannot be declared to be superior or universally applicable. A study carried out by Milligan [101], indicated that single-linkage methods produce better results when outliers are present in the dataset, whereas average-linkage methods are better to assess the true cluster structure in the presence of noisy data points. Along with the algorithm of choice, a decision also has to be made regarding the metric to be used to measure the similarity or distance between clusters.

2.4.3 Hierarchical clustering using ScrumPy

The versatility of ScrumPy permits the implementation of any clustering algorithm or distance measure that may be of interest to the user. ScrumPy can be used to carry out hierarchical clustering in conjunction with a choice of popular distance measures such as the euclidean distance between vectors. All dynamic matrices such as an EMs stoichiometry or reaction matrix (Appendix A) generated using ScrumPy provide a method to calculate a distance matrix based on row attributes:

```
>>> DistFunc = ScrumPy.Util.Seq.EucDist
>>> rdiff = mtx.RowDiffMtx(fun=DistFunc,Conv=float)
```

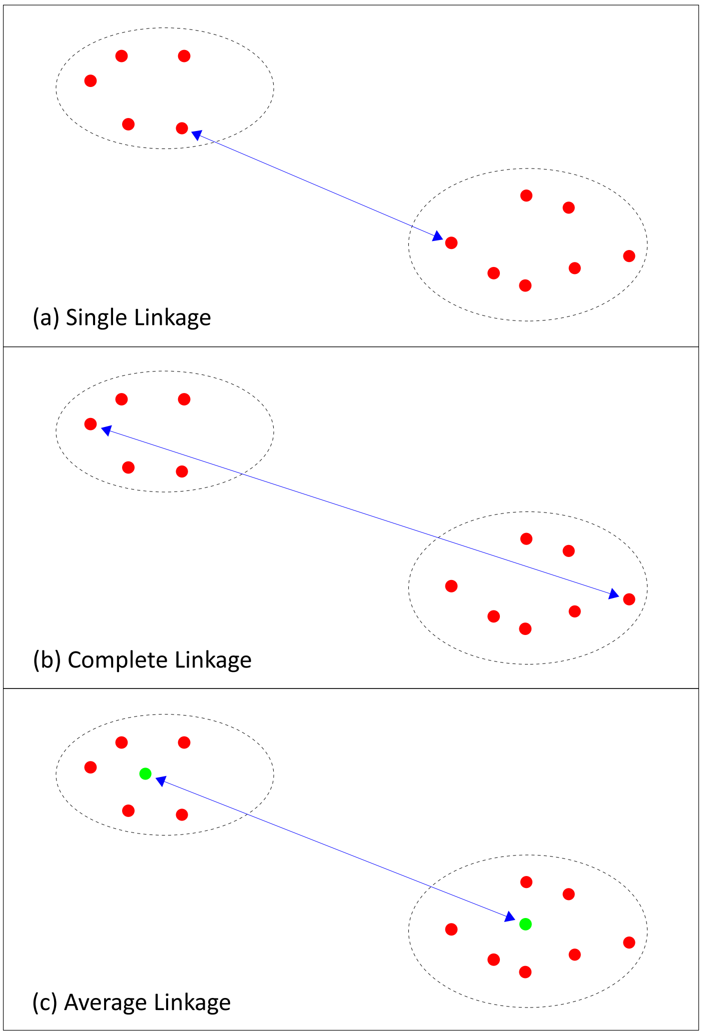


Figure 2.6 – Illustration to show how the distance (or similarity) between clusters is defined. (*a*) Single linkage measures the shortest distance between objects in different clusters, (*b*) complete linkage measures the largest distance between objects in different clusters and, (*c*) average linkage measures the mean of all the objects in each cluster for comparison.

where `RowDiffMtx` is a method that compares pairs of rows from `mtx` and returns a distance matrix (`rdiff`) with elements (`ArbRat` by default; see Appendix A) that indicate how similar one row is to another, using a measure of choice (i.e in this case `DistFunc` represents the euclidean distance). Other distance measures or element types may be employed by simply changing the arguments passed to `RowDiffMtx`. Subsequently, a Python string representation of the hierarchical tree can be calculated from `rdiff` and returned in Newick format[1]:

```
>>> newick = rdiff.ToNJTree()
>>> print newick
'(B:5,(A:1,C:1):3,(D,E:0):3);'
```

[1] standard representation of a graph theoretical tree in text format using parentheses and commas.

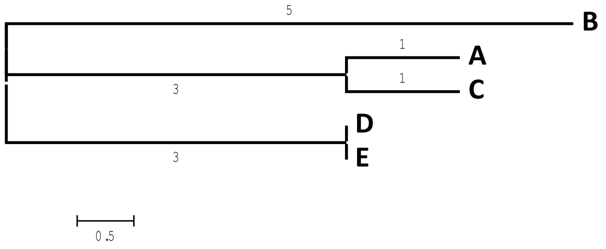


Figure 2.7 – A test dendrogram generated in the Newick format. The scale below represents a measure of distance per unit branch length.

As shown in Figure 2.7, saving the Newick tree from the ScrumPy output in a file will enable the user to subsequently view, manipulate and label the tree using external phylogenetic tree viewing programs that have been implemented for this purpose, including NJPlot† and MEGA† (software of choice). Other popular multi-purpose phylogenetic software packages include the freely-available PHYLIP† and SplitsTree† and commercial PAUP†. As introduced in the next section, validation of hierarchical clustering methods with application to EMA will be carried out on a small model of yeast anaerobic central metabolism. Subsequently, the results obtained will be presented through various visualisation methods for use in the final discussion.

2.5 Clustering elementary modes using a test yeast model

To illustrate the usefulness of hierarchical clustering methods for application to EMs datasets a small model of yeast anaerobic central metabolism was reconstructed for the production of ethanol and/or glycerol via glycolytic and pentose phosphate pathway reactions. The model was built using yeast specific metabolic pathway maps† accessed from the BioCyc family of databases. Since there is no new biochemical knowledge to be gained for this relatively small and well documented metabolic model, carbon entities were denoted by abstract metabolite identifiers to focus on the clustering results (Figure 2.8 and related '*.spy*' model in Appendix B). Before the EMs calculation the initial reactions in the model were replaced by the overall stoichiometry of the enzyme subset in which they participate. This has no effect on the resulting EMs dataset, however, at the same time reducing the complexity and increasing the interpretability of the model (Section 1.6.3.1). The final model contained a total of 7 reactions involving 4 internal metabolites and 4 external metabolites. As shown in Figure 2.8, aside from carbon dioxide (X_2) exchange, 3 transport reactions were defined to highlight the input of glucose (X_1) and output of glycerol (X_3) and ethanol (X_4).

The stoichiometric coefficients have not been shown on the graphical representation for the model (Figure 2.8). Even for simple models the graphical layout of the network relies on a compromise that aims to reduce the overlap in spatial placement to increase visual interpretation. Furthermore, the interconnectivity between nodes makes it very difficult to include additional information such as stoichiometric coefficients. Owing to this, on inspection of Figure 2.8 it seems that there is a stoichiometric inconsistency between R_2 and R_5, since $S_1 \longrightarrow S_2$ and $S_1 \longrightarrow S_2 + S_4 + X_2$, respectively, are not valid together. The actual reaction equation for R_5 is 3 $S_1 \longrightarrow 2\ S_2 + S_4 + 3\ X_2$, and all metabolites in the remainder of the reactions have a stoichiometric coefficient of 1. Knowledge of this stoichiometric information will help in the interpretation of the 9 modes yielded from the EMA (Figure 2.9) since it indicates to which proportion different reactions must be used together to balance internal metabolites, and thereafter, for the determination of the net production or consumption of external metabolites.

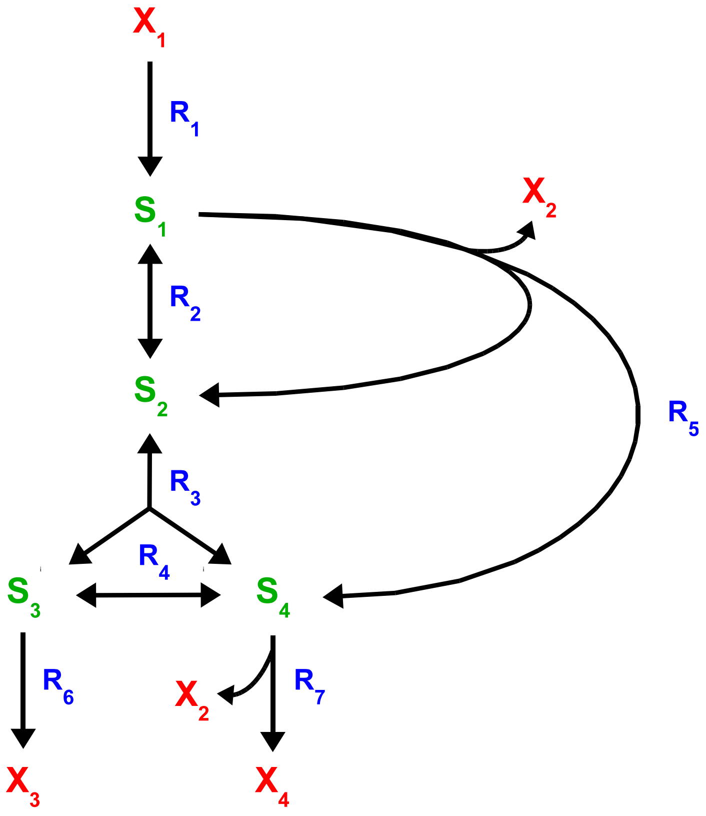


Figure 2.8 – Reaction schematic of the test yeast model. Reaction reversibility is as indicated in the diagram and internal and external metabolites have been highlighted in green and red, respectively.

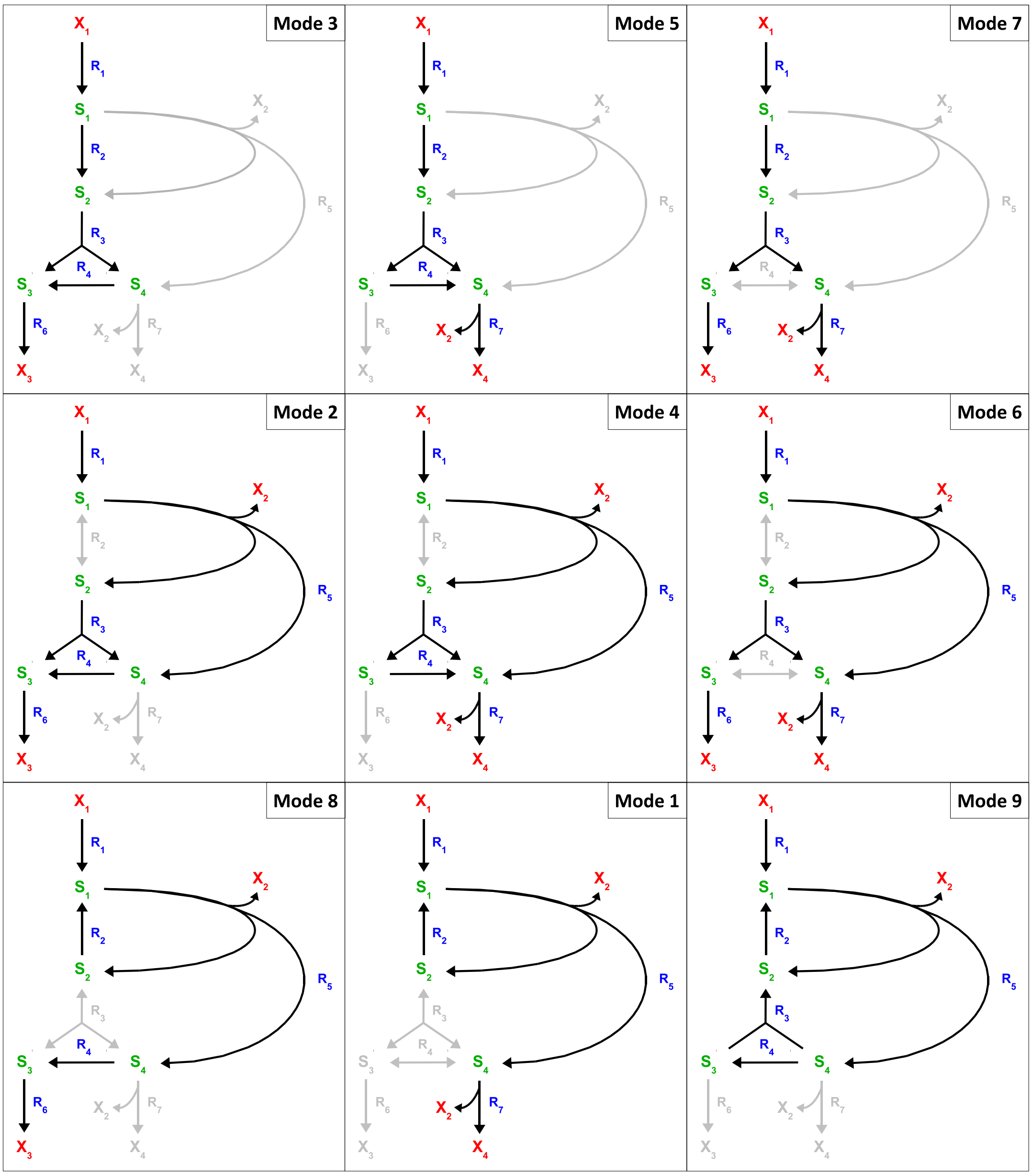


Figure 2.9 – Illustration to show the 9 EMs generated from the test yeast model in Figure 2.8. All modes are irreversible and reaction directionality has been indicated on the graphical representation for individual modes. Modes have been ordered according to similarities in pathway usage.

2.5.1 Visualisation of clustering results

All hierarchical trees were generated using an agglomerative algorithm implemented in ScrumPy using angle as a distance measure to make use of its ability to cluster modes with similar overall profiles. Subsequently, two methods for labelling the branch information on the generated dendrograms will be compared. Initially, the conventional dendrographic visualisation with textual branch descriptions relating to the clustered data will be presented followed by a coloured visualisation method representing the original data matrix as a substitute for the textual information.

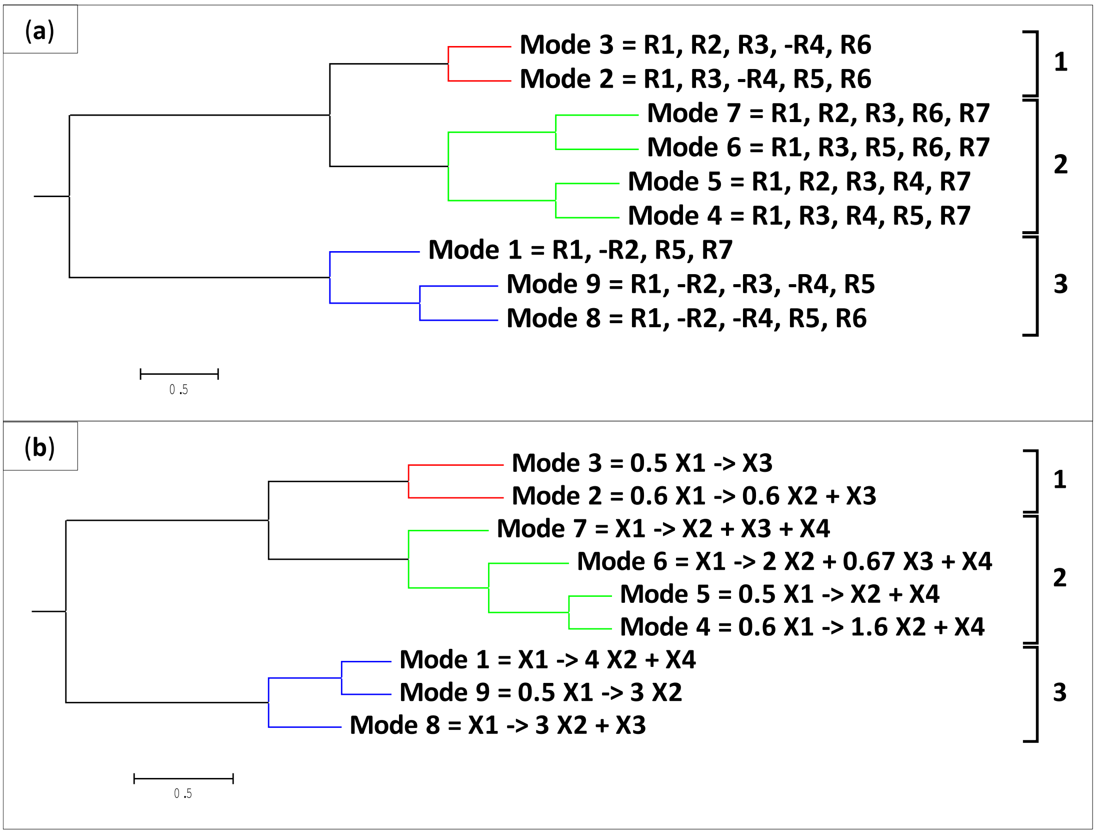

(a)
Mode 3 = R1, R2, R3, -R4, R6
Mode 2 = R1, R3, -R4, R5, R6
Mode 7 = R1, R2, R3, R6, R7
Mode 6 = R1, R3, R5, R6, R7
Mode 5 = R1, R2, R3, R4, R7
Mode 4 = R1, R3, R4, R5, R7
Mode 1 = R1, -R2, R5, R7
Mode 9 = R1, -R2, -R3, -R4, R5
Mode 8 = R1, -R2, -R4, R5, R6
1
2
3
0.5
(b)
Mode 3 = 0.5 X1 -> X3
Mode 2 = 0.6 X1 -> 0.6 X2 + X3
Mode 7 = X1 -> X2 + X3 + X4
Mode 6 = X1 -> 2 X2 + 0.67 X3 + X4
Mode 5 = 0.5 X1 -> X2 + X4
Mode 4 = 0.6 X1 -> 1.6 X2 + X4
Mode 1 = X1 -> 4 X2 + X4
Mode 9 = 0.5 X1 -> 3 X2
Mode 8 = X1 -> 3 X2 + X3
1
2
3
0.5

Figure 2.10 – Dendrogram to show the EMs from the test yeast model whereby modes were clustered by angle according to their (*a*) reaction usage and, (*b*) net external metabolite usage. Coloured branches highlight clusters containing the same modes in both (*a*) and (*b*).

2.5.1.1 Dendrograms

An EM can be best described in terms of its reaction profile and/or its net stoichiometry. Therefore, two dendrograms were obtained by clustering the entire EMs reaction (Figure 2.10(*a*)) and stoichiometry matrices (Figure 2.10(*b*)). The ScrumPy Newick format output representing each dendrogram was written to file and then imported into MEGA for subsequent visualisation and modification.

2.5.1.2 Matrix visualisation

Eisen *et al.* [88] introduced a visualisation method whereby a dendrographic representation of their gene expression data is appended to a colour coded matrix[2] to indicate the clustered relationships among genes. Unchanged genes were coloured black (=0) and increasing intensities of red and green were used to indicate increasing ratios of overexpressed (>0) and underexpressed genes (<0), respectively. An efficient dimension reduction was achieved using this method to identify patterns of interest amongst their high dimensional data. Similarly, it was of interest

[2] a reordered copy of the primary data table used to generate the clustering output.

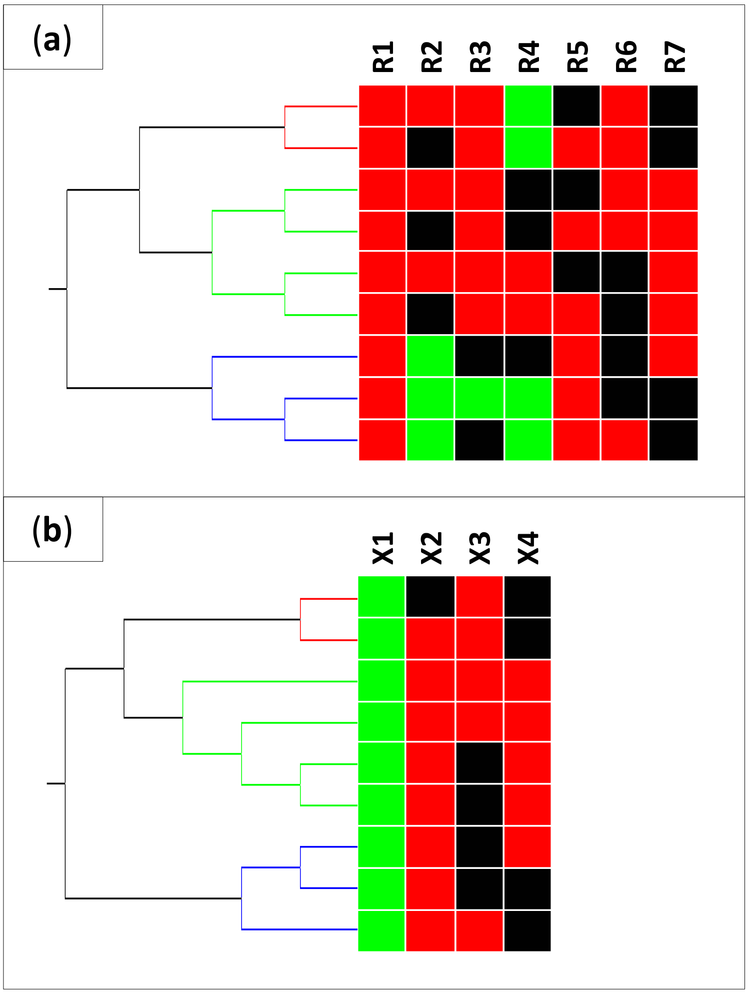


Figure 2.11 – Dendrogram and matrix visualisation for the EMs from the test yeast model, as generated in Figure 2.10, however, branch information in (*a*) and (*b*) has been replaced with the corresponding EM reaction and stoichiometry matrix, respectively. Matrices have been coloured according to coefficient value i.e. =0 (black), <0 (green) and >0 (red).

to determine whether a dendrogram-coloured matrix graphic would be useful to interpret clustered EMs datasets. The original EMs matrix used to generate the clustering output can be reordered to reflect the structure in the dendrogram and coloured according to the coefficients in the matrix i.e. =0 (black), <0 (green) and >0 (red).

2.6 Discussion

The large number of modes yielded from an EMA are manually interpreted in an individual fashion and are usually chosen based on their participation in the objective process that was defined when building the model (e.g. production of ethanol from glucose). Therefore, of particular interest to this project was the development of more globally applied and informative post-analytical methods that can be used to interpret EMs datasets. Hierarchical clustering techniques were chosen for their ability to group and, subsequently, investigate both the large- and small-scale features of the dataset.

As highlighted in Figure 2.9, 9 modes were found for the test yeast anaerobic model described in Section 2.5. Such a side-by-side graphical display is possible for the entire EM dataset only because the number of generated modes is small. However, for larger EMs datasets it simply not possible to graphically display and visually inspect all the pathway variations that may arise, although, this may be possible for a handful of modes of interest. Nevertheless, in the first instance, it is worthwhile clustering the output from the EMA to reveal groups of modes that are using reactions in a similar fashion. Thereafter it is possible to select and visualise pathway representatives of individual clusters. As shown in Figure 2.10(a), clustering by reaction profile produces three distinct clusters:

- **Cluster 1**
  Contains two pathway variations for the production of external metabolite X_3 from X_1 or from X_1 and X_2. The former and latter stoichiometries arise due to the lack of usage and usage of reaction R_5, respectively.

- **Cluster 2**
  Similar to cluster 1 the modes in this cluster are paired into two groups according to their usage of R_5. The first group (Modes 6 and 7) indicate routes for the production of X_2, X_3 and X_4 from X_1 whereas the second group (Modes 4 and 5) highlights the production of X_2 and X_4 from X_1.

- **Cluster 3**
  All the modes in this cluster utilise R_2 in the backwards direction to recycle internal metabolite S_1 whilst using R_5 in conjunction with the absence or backwards usage of R_3. Three separate routes exist for the production of X_1 and X_2, X_3 and X_2 or just X_2.

With the exception of futile cycles[3] (Section 1.6.3.1), both the stoichiometric and reaction information are relative for different modes since the net usage of metabolites is determined by the reactions which are used to consume and produce them. This cannot be demonstrated better than by comparing the clusters obtained from the dendrograms generated by clustering in terms of reaction profile (Figure 2.10(a)) and net stoichiometry (Figure 2.10(b)). However, for larger models the vast number of reaction or stoichiometric combinations that may occur implies that the pair of dendrograms obtained by clustering according to reaction profile and net stoichiometry may not be comparable. This simply indicates

[3] modes that balance internal metabolites without a net external metabolite usage.

that neither clustering method is superior and are most informative when used alongside each other.

The coloured matrix representations of Figure 2.10, as shown in Figure 2.11 can be used to highlight the importance of replacing purely textual descriptions of EMs on the branches of the dendrogram with a visualisation that indicates their overall usage of reactions or external metabolites, respectively. Using Figure 2.11(a) as an example, at first sight, it becomes immediately apparent that:

- all modes use R_1 in a positive direction.
- the modes in the first two clusters differ in the alternating and opposite use and disuse of R_2 and R_5.
- all the modes in the final cluster use R_2 in a negative direction and R_5 in a positive direction.

Although the observations listed above were also made whilst discussing the dendrogram in Figure 2.10(a) earlier in the section, even for a model of this size, it is much easier to come to these conclusions whilst looking at a visualisation as opposed to comparing textual descriptions. Additionally, global variations for the usage of reactions between clusters of modes becomes more apparent (e.g. reaction(s) used in a positive direction in one cluster of modes and in a negative direction in all other clusters). Following on from this, local variations in individual clusters of interest can be focused upon and may be used to further interpret and characterise the modes. Therefore, in summary, the clustering and subsequent visualisation methods as applied to the EMs dataset from the test yeast anaerobic model have a number of potential benefits for:

- grouping EMs according to their similarities in reaction or stoichiometric profiles.
- enabling the researcher to acquire a global view of the functioning of the modes. This is advantageous for obtaining an unbiased representation of all the modes generated. Groups of EMs with similar net metabolic conversions can be investigated together to reduce the time needed for post-analytical investigations.
- interpreting the way in which different reactions are being utilised by groups of modes.
- an effective dimension reduction by using a matrix visualisation and dendrogram combination to aid in both the global and local interpretation of the modes in the tree.

Taking these advantages into account, hierarchical clustering techniques with application to EMA will also be carried out in the next chapter on a more extensive model of central metabolic pathways. Consequently, this will be used as a secondary validation as to their usefulness for the interpretation of EM datasets and, ultimately, to a gain in biochemical knowledge for the model in question.

CHAPTER 3

S. erythraea Model of Central Metabolism

3.1 Introduction

Saccharopolyspora erythraea [102] is a gram-positive, soil-dwelling bacterium that is used industrially to produce the clinically important polyketide antibiotic erythromycin. Erythromycin is a potent inhibitor of ribosomal protein synthesis in bacteria and has been the basis of much interest in the drug industry [103]. In microbial metabolism, the flux through pathways involving secondary metabolites is insignificant when compared to total carbon metabolism. To date, through fermentation experiments with *S. erythraea* it has been reported that during the growth phase 2-oxoglutarate is exported into the extracellular environment whereas during the stationary phase this process is reversed whilst producing erythromycin (University College London, *personal communication*). The exact reason for this behaviour is not yet known. However, analogous to the behaviour of lactic acid producing Lactobacilli, the acidic media conditions created by transporting 2-oxoglutarate out of the cell may reduce competitive growth from other bacteria. Due to glucose exhaustion upon entering the stationary phase, the 2-oxoglutarate may then be taken back into the cell as a replacement carbon source. Therefore, it would be of interest to investigate the flow of carbon through primary metabolites such as 2-oxoglutarate.

3.2 Model reconstruction

A structural model was reconstructed manually using MetaCyc pathway map visualisations as a reference for central metabolic pathways ('*.spy*' model in Appendix C). It was built to investigate the production of 2-oxoglutarate from glucose, exclusively via central metabolism. At this point it is worth noting that the model definition was not extrapolated from the genome sequence of *S. erythraea* but from its predicted biochemical capabilities. The final version of the model contained 34 reactions and 45 metabolites. The pathways included in the model are illustrated in Figure 3.1, and their corresponding reaction sets are listed below:

- **Glycolysis**
  Responsible for the net conversion of glucose into pyruvate (R_1–R_{11}). Pyruvate is subsequently fed into the pyruvate dehydrogenase complex (R_{22}) which makes acetyl-CoA available to the TCA cycle.

- **Pentose phosphate pathway**
  An alternative pathway to glycolysis for oxidising glucose, in this case coupled with NADPH synthesis. Consists of oxidative (R_{12} and R_{13}) and non-oxidative branches (R_{14}–R_{19}).

- **Entner-Doudoroff pathway**
  Uses the oxidative pentose phosphate pathway along with two reactions of its own (R_{20} and R_{21}) for the phosphorylation and subsequent cleavage by an aldolase of 6-carbon sugars into two 3-carbon intermediates (i.e. pyruvate and glyceraldehyde-3-phosphate).

- **TCA cycle**
  A component of central metabolism that operates under various conditions, the reactions of the TCA cycle yield three important precursor metabolites, 2-oxoglutarate, succinyl-CoA, and oxaloacetate along with energy and reducing equivalents (R_{23}–R_{32}).

- **Glyoxylate cycle**
  Bypasses reactions of the TCA cycle which evolve CO_2 and conserves 4-carbon compounds for biosynthesis (R_{33} and R_{34}).

Metabolites were declared external (Section 1.2.1) based on the following criteria. Firstly, two transport reactions (R_1 and R_{27}) were included in the model for the metabolites involved in the objective process that was defined when building the model (i.e. exchange of glucose and 2-oxoglutarate). Other carbon exchange metabolites including CO_2 and HCO_3^- were also made external. Secondly, those metabolites that are likely to be in constant exchange with the extracellular environment in living cells (e.g. water and protons). Thirdly, cofactor metabolites such as ADP/ATP and $NADP^+$/NADPH responsible for energy and reducing power, respectively. With the use of the relevant interconverting reactions, cofactor metabolites would usually be conserved at steady state if kept internal. As model size increases, cofactor metabolites can be made external to decrease the connectivity within the model to aid analytical procedures such as EMA (Section 1.6.3.1). However, for the purpose of this investigation they were defined as external in order to directly derive net energy and reducing yields from the net stoichiometry of the EMs to be generated later.

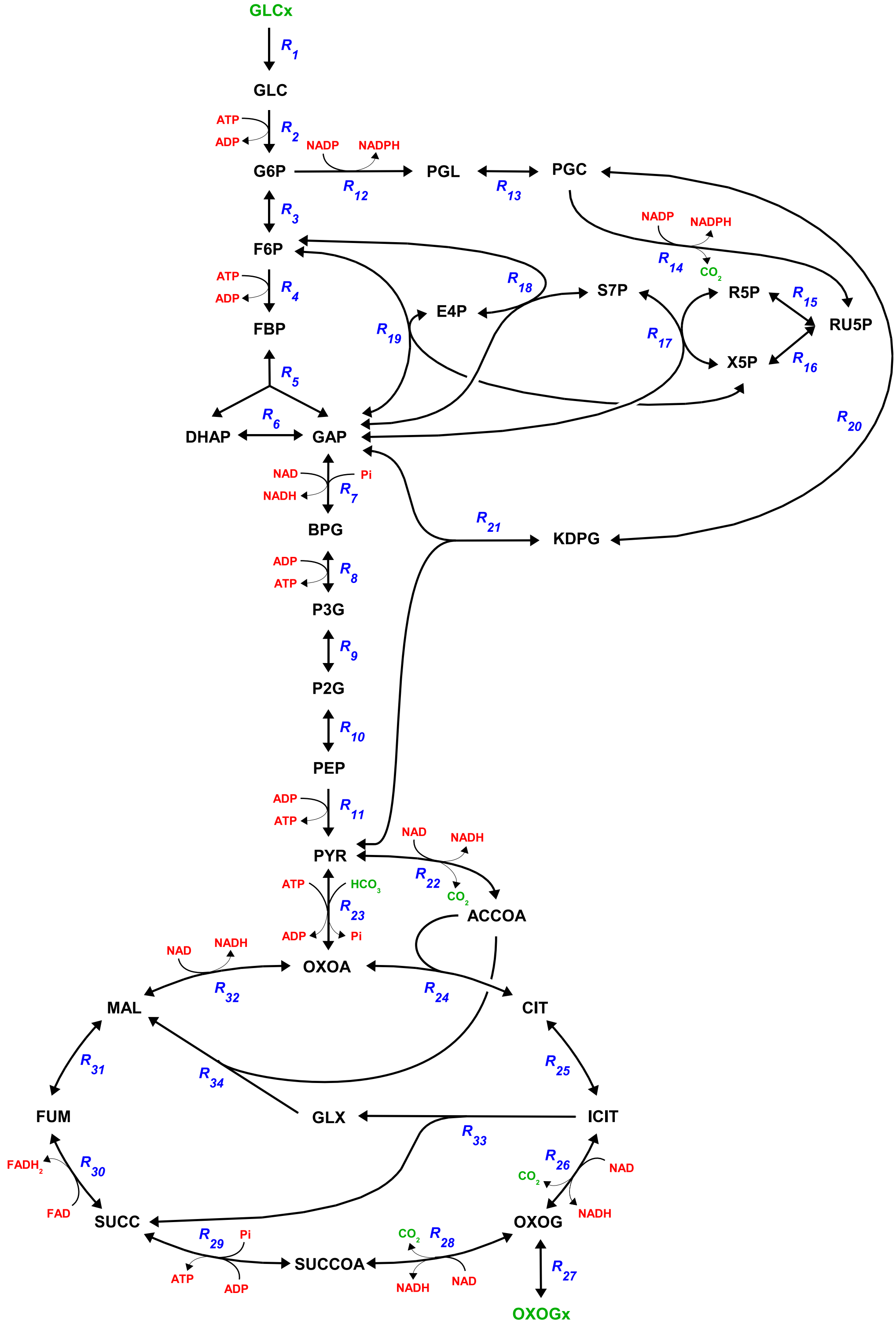


Figure 3.1 – Reaction schematic of the minimal 2-oxoglutarate model. All external metabolites except for water and proton have been included in the diagram. Carbon exchange metabolites and energy and reducing equivalents have been highlighted in green and red, respectively. For purposes of clarity, coenzyme A is the only internal metabolite that has been excluded and is consumed in reactions 22 and 28, and produced in reactions 24, 29 and 34. As indicated in the figure, the metabolites in all the reactions in the model have a stoichiometric coefficient of 1. See List of Abbreviations for metabolite abbreviations and Appendix D for a list of enzymes as correlated to the reactions in the figure.

3.3 General model analysis

As would be expected from a hand-built model of this size there are no orphan metabolites or dead reactions, and all reactions were atomically balanced. Enzyme subsets analysis revealed 10 subsets containing more than one reaction and 6 involving single reactions (Figure 3.2). Table 3.1 shows the obtained list of enzyme subsets along with their relationship to the individual reactions in the model (Figure 3.1) and, furthermore, the pathways they are involved in. The largest subset was found to comprise six reactions forming the non-oxidative component of the pentose phosphate pathway. The reactions that form the oxidative pentose phosphate pathway, Entner-Doudoroff pathway and glyoxylate bypass were also separated into subsets of their own. The lower linear part of glycolysis formed a single subset, however, the upper glycolytic reactions were found to consist of one single and two multiple reaction subsets due to the pathway branching that occurs for the entry of glucose-6-phosphate into the pentose phosphate pathway. The TCA cycle was found to have 7 subsets due to the branching that occurs from the entry and exit of metabolites involved in the glyoxylate cycle. The biological significance of these results will be discussed later.

A total of 55 EMs were found using the EMs algorithm. The visualisation techniques in the next section will be used to further interrogate and classify the EMs dataset.

3.4 Visualisation of clustering results

A logical way of viewing complex datasets is first to scan and survey the large-scale features and then to converge on the interesting details. Initially, a series of dendrograms will be generated according to differing criteria of interest within the entire EMs dataset. Thereafter, the matrix visualisation method introduced in Section 2.5.1.2 will be used for further interrogation of the smaller scale features. All hierarchical trees were generated using an agglomerative algorithm implemented in ScrumPy using angle as the distance measure (Section 2.4.3). The MEGA clustering software was subsequently used for visualisation and modification of the dendrograms which were exported from ScrumPy in Newick format.

3.4.1 Dendrograms

A pair of dendrograms were obtained by clustering the entire EMs reaction (Figure 3.3) and stoichiometry matrices (Figure 3.4). Two additional dendrograms were generated by clustering a condensed EMs stoichiometry matrix according to carbon

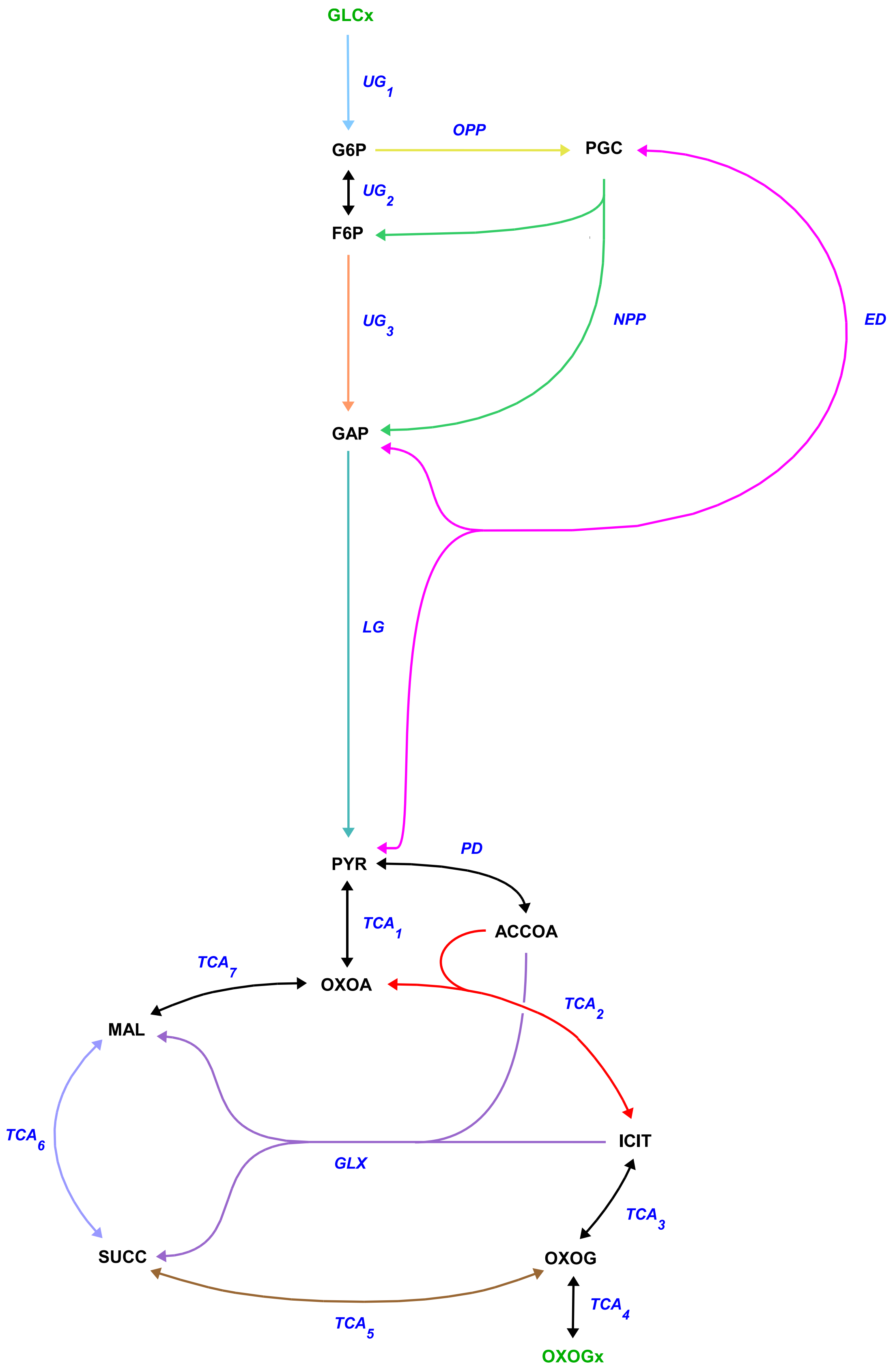


Figure 3.2 – Reaction schematic of the minimal 2-oxoglutarate model with enzyme subsets highlighted. Reactions with black reversibility arrows indicate reactions that are in a subset of their own. See Table 3.1 for the list of pathways as correlated to the subset abbreviations used in the illustration and Figure 3.1 to derive the net external metabolite usage for each subset.

Pathway	Reaction Subscript	Pathway Subset Abbreviation
Upper glycolysis	1 2	UG1
	3	UG2
	4 5 6	UG3
Lower glycolysis	7 8 9 10 11	LG
Oxidative pentose phosphate	12 13	OPP
Non-oxidative pentose phosphate	14 15 16 17 18 19	NPP
Entner-Doudoroff	20 21	ED
Pyruvate dehydrogenase	22	PD
Citric acid cycle	23	TCA1
	24 25	TCA2
	26	TCA3
	27	TCA4
	28 29	TCA5
	30 31	TCA6
	32	TCA7
Glyoxylate cycle	33 34	GLX

Table 3.1 – Table to show the enzyme subsets for the minimal 2-oxoglutarate model as related to individual pathways. See Figure 3.1 for reaction subscripts and Figure 3.2 for a graphical representation of the subsets. The pathway subset abbreviations will be used in the clustering results presented in the next section.

entities (Figure 3.5) and energy and reducing equivalents (Figure 3.6), respectively. On further inspection of Figure 3.5, the modes were classified into five distinct groups according to their net carbon stoichiometry, and as indicated in the list below each group was given a different branch colouration for ease of identification in all the generated dendrograms, those that:

1. produce 2-oxoglutarate from glucose (24).
2. consume 2-oxoglutarate to produce carbon dioxide and/or bicarbonate (13).
3. consume glucose to produce carbon dioxide and/or bicarbonate (11).
4. others that do not involve glucose or 2-oxoglutarate (5).
5. produce 2-oxoglutarate from carbon dioxide and/or bicarbonate (2).

3.4.2 Matrix visualisation

As introduced in the previous chapter (Section 2.5.1.2), a dendrogram-matrix visualisation method will be employed to further investigate the clustering results. It was applied firstly, to all the EMs from the minimal model according to their net external metabolite usage of carbon entities (Figure 3.7); secondly, to the modes that produce 2-oxoglutarate from glucose according to their reaction usage (Figure 3.8), and thirdly, to the 8 modes that produce 2-oxoglutarate from glucose without carbon dioxide fixation according to their reaction usage (Figure 3.9).

3.5 Discussion

The results from the enzyme subsets analysis were very informative. Since an enzyme subset is a group of reactions operating at steady state, therefore, an EM also calculated at steady state is simply a combination of enzyme subsets with a net balance of internal metabolites. As shown in Figure 3.2, the classical pathways included in the model were found to form distinct subsets which were useful for further analysis of the EMs dataset. It is much easier to decipher the biological meaning of an EM if its component reactions are converted into their corresponding enzyme subsets, and described in terms of the pathways they operate in.

In vivo it would be expected that all the reactions within the pathways mentioned (i.e as separated into enzyme subsets) would be expected to be active at the same time. A study was recently carried out within our group to reinforce the link between the co-regulated gene expression of groups of enzymes according to the metabolic requirements in *E. coli* [43]. An important outcome from the

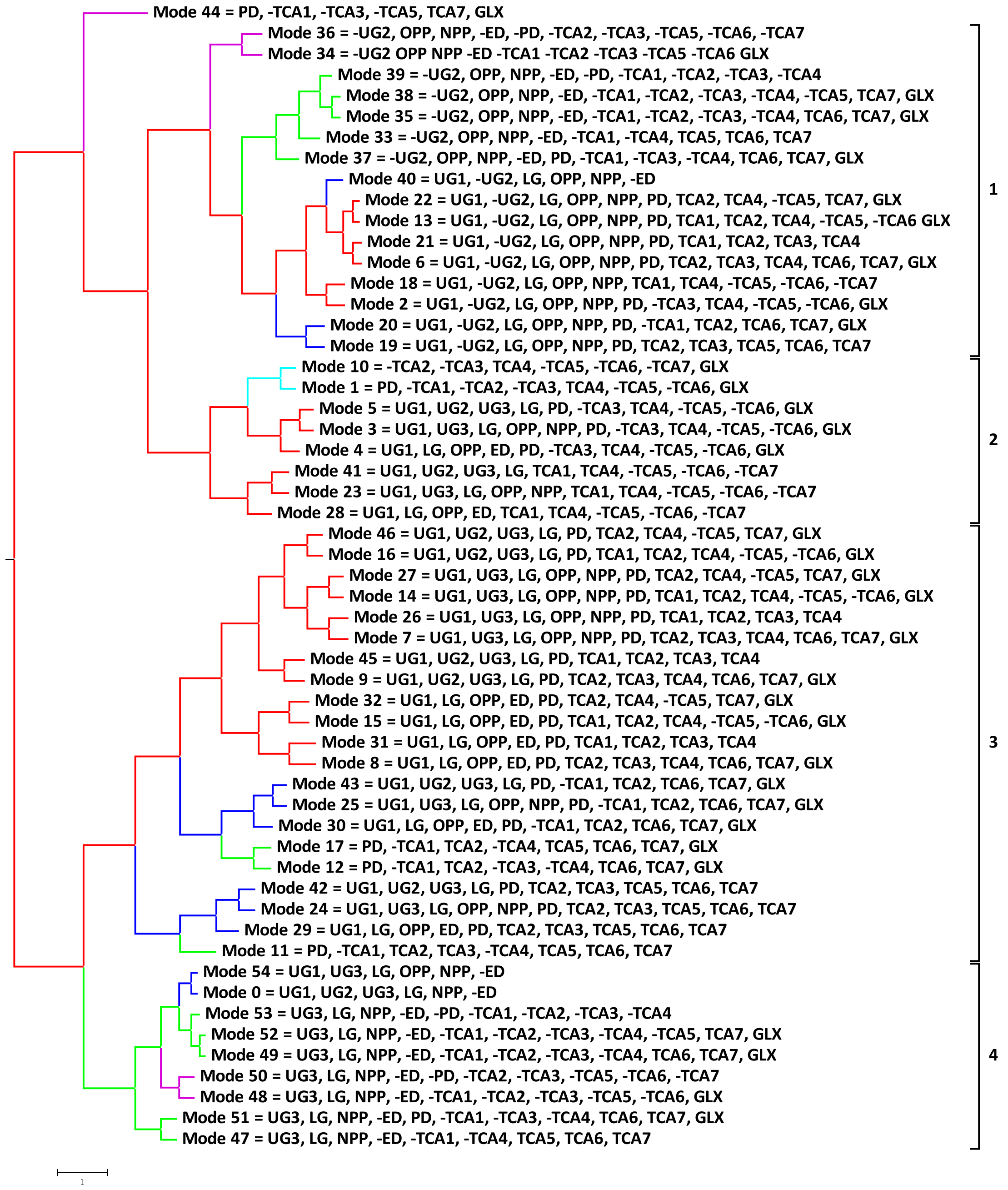

Mode 44 = PD, -TCA1, -TCA3, -TCA5, TCA7, GLX
Mode 36 = -UG2, OPP, NPP, -ED, -PD, -TCA2, -TCA3, -TCA5, -TCA6, -TCA7
Mode 34 = -UG2 OPP NPP -ED -TCA1 -TCA2 -TCA3 -TCA5 -TCA6 GLX
Mode 39 = -UG2, OPP, NPP, -ED, -PD, -TCA1, -TCA2, -TCA3, -TCA4
Mode 38 = -UG2, OPP, NPP, -ED, -TCA1, -TCA2, -TCA3, -TCA4, -TCA5, TCA7, GLX
Mode 35 = -UG2, OPP, NPP, -ED, -TCA1, -TCA2, -TCA3, -TCA4, TCA6, TCA7, GLX
Mode 33 = -UG2, OPP, NPP, -ED, -TCA1, -TCA4, TCA5, TCA6, TCA7
Mode 37 = -UG2, OPP, NPP, -ED, PD, -TCA1, -TCA3, -TCA4, TCA6, TCA7, GLX
Mode 40 = UG1, -UG2, LG, OPP, NPP, -ED
Mode 22 = UG1, -UG2, LG, OPP, NPP, PD, TCA2, TCA4, -TCA5, TCA7, GLX
Mode 13 = UG1, -UG2, LG, OPP, NPP, PD, TCA1, TCA2, TCA4, -TCA5, -TCA6 GLX
Mode 21 = UG1, -UG2, LG, OPP, NPP, PD, TCA1, TCA2, TCA3, TCA4
Mode 6 = UG1, -UG2, LG, OPP, NPP, PD, TCA2, TCA3, TCA4, TCA6, TCA7, GLX
Mode 18 = UG1, -UG2, LG, OPP, NPP, TCA1, TCA4, -TCA5, -TCA6, -TCA7
Mode 2 = UG1, -UG2, LG, OPP, NPP, PD, -TCA3, TCA4, -TCA5, -TCA6, GLX
Mode 20 = UG1, -UG2, LG, OPP, NPP, PD, -TCA1, TCA2, TCA6, TCA7, GLX
Mode 19 = UG1, -UG2, LG, OPP, NPP, PD, TCA2, TCA3, TCA5, TCA6, TCA7
Mode 10 = -TCA2, -TCA3, TCA4, -TCA5, -TCA6, -TCA7, GLX
Mode 1 = PD, -TCA1, -TCA2, -TCA3, TCA4, -TCA5, -TCA6, GLX
Mode 5 = UG1, UG2, UG3, LG, PD, -TCA3, TCA4, -TCA5, -TCA6, GLX
Mode 3 = UG1, UG3, LG, OPP, NPP, PD, -TCA3, TCA4, -TCA5, -TCA6, GLX
Mode 4 = UG1, LG, OPP, ED, PD, -TCA3, TCA4, -TCA5, -TCA6, GLX
Mode 41 = UG1, UG2, UG3, LG, TCA1, TCA4, -TCA5, -TCA6, -TCA7
Mode 23 = UG1, UG3, LG, OPP, NPP, TCA1, TCA4, -TCA5, -TCA6, -TCA7
Mode 28 = UG1, LG, OPP, ED, TCA1, TCA4, -TCA5, -TCA6, -TCA7
Mode 46 = UG1, UG2, UG3, LG, PD, TCA2, TCA4, -TCA5, TCA7, GLX
Mode 16 = UG1, UG2, UG3, LG, PD, TCA1, TCA2, TCA4, -TCA5, -TCA6, GLX
Mode 27 = UG1, UG3, LG, OPP, NPP, PD, TCA2, TCA4, -TCA5, TCA7, GLX
Mode 14 = UG1, UG3, LG, OPP, NPP, PD, TCA1, TCA2, TCA4, -TCA5, -TCA6, GLX
Mode 26 = UG1, UG3, LG, OPP, NPP, PD, TCA1, TCA2, TCA3, TCA4
Mode 7 = UG1, UG3, LG, OPP, NPP, PD, TCA2, TCA3, TCA4, TCA6, TCA7, GLX
Mode 45 = UG1, UG2, UG3, LG, PD, TCA1, TCA2, TCA3, TCA4
Mode 9 = UG1, UG2, UG3, LG, PD, TCA2, TCA3, TCA4, TCA6, TCA7, GLX
Mode 32 = UG1, LG, OPP, ED, PD, TCA2, TCA4, -TCA5, TCA7, GLX
Mode 15 = UG1, LG, OPP, ED, PD, TCA1, TCA2, TCA4, -TCA5, -TCA6, GLX
Mode 31 = UG1, LG, OPP, ED, PD, TCA1, TCA2, TCA3, TCA4
Mode 8 = UG1, LG, OPP, ED, PD, TCA2, TCA3, TCA4, TCA6, TCA7, GLX
Mode 43 = UG1, UG2, UG3, LG, PD, -TCA1, TCA2, TCA6, TCA7, GLX
Mode 25 = UG1, UG3, LG, OPP, NPP, PD, -TCA1, TCA2, TCA6, TCA7, GLX
Mode 30 = UG1, LG, OPP, ED, PD, -TCA1, TCA2, TCA6, TCA7, GLX
Mode 17 = PD, -TCA1, TCA2, -TCA4, TCA5, TCA6, TCA7, GLX
Mode 12 = PD, -TCA1, TCA2, -TCA3, -TCA4, TCA6, TCA7, GLX
Mode 42 = UG1, UG2, UG3, LG, PD, TCA2, TCA3, TCA5, TCA6, TCA7
Mode 24 = UG1, UG3, LG, OPP, NPP, PD, TCA2, TCA3, TCA5, TCA6, TCA7
Mode 29 = UG1, LG, OPP, ED, PD, TCA2, TCA3, TCA5, TCA6, TCA7
Mode 11 = PD, -TCA1, TCA2, TCA3, -TCA4, TCA5, TCA6, TCA7
Mode 54 = UG1, UG3, LG, OPP, NPP, -ED
Mode 0 = UG1, UG2, UG3, LG, NPP, -ED
Mode 53 = UG3, LG, NPP, -ED, -PD, -TCA1, -TCA2, -TCA3, -TCA4
Mode 52 = UG3, LG, NPP, -ED, -TCA1, -TCA2, -TCA3, -TCA4, -TCA5, TCA7, GLX
Mode 49 = UG3, LG, NPP, -ED, -TCA1, -TCA2, -TCA3, -TCA4, TCA6, TCA7, GLX
Mode 50 = UG3, LG, NPP, -ED, -PD, -TCA2, -TCA3, -TCA5, -TCA6, -TCA7
Mode 48 = UG3, LG, NPP, -ED, -TCA1, -TCA2, -TCA3, -TCA5, -TCA6, GLX
Mode 51 = UG3, LG, NPP, -ED, PD, -TCA1, -TCA3, -TCA4, TCA6, TCA7, GLX
Mode 47 = UG3, LG, NPP, -ED, -TCA1, -TCA4, TCA5, TCA6, TCA7
1
2
3
4
1

Figure 3.3 – Dendrogram to show all the EMs from the minimal 2-oxoglutarate model clustered by angle according to their reaction usage, and collapsed into their subsets. See Table 3.1 for pathway subset abbreviations. Branch colouration indicates differing net carbon stoichiometries as explained in Section 3.4.1. Flux values have not been included for purposes of clarity. Individual clusters have been numbered for interpretative purposes and will be referred to in the discussion section.

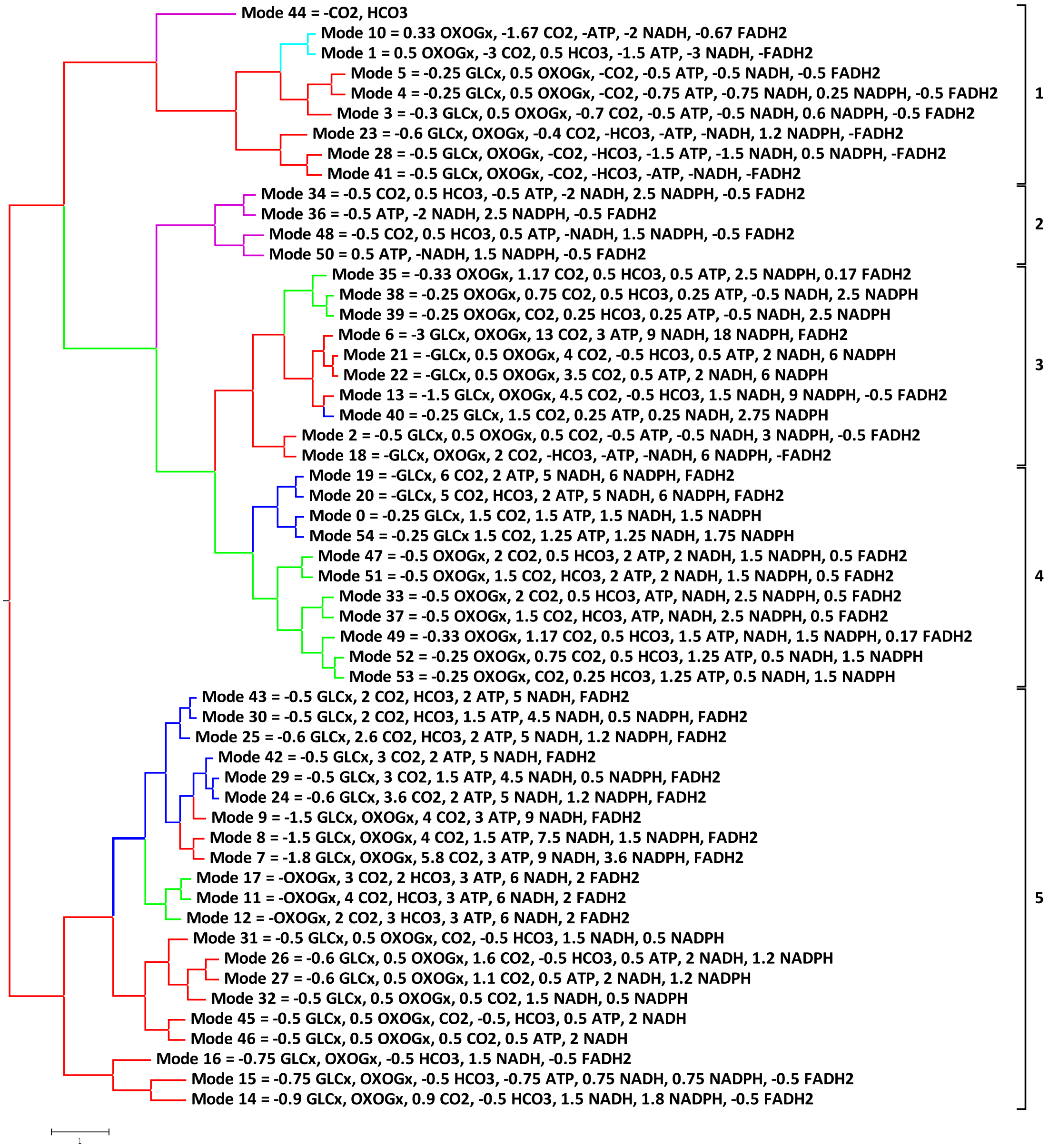

Mode 44 = -CO2, HCO3
Mode 10 = 0.33 OXOGx, -1.67 CO2, -ATP, -2 NADH, -0.67 FADH2
Mode 1 = 0.5 OXOGx, -3 CO2, 0.5 HCO3, -1.5 ATP, -3 NADH, -FADH2
Mode 5 = -0.25 GLCx, 0.5 OXOGx, -CO2, -0.5 ATP, -0.5 NADH, -0.5 FADH2
Mode 4 = -0.25 GLCx, 0.5 OXOGx, -CO2, -0.75 ATP, -0.75 NADH, 0.25 NADPH, -0.5 FADH2
Mode 3 = -0.3 GLCx, 0.5 OXOGx, -0.7 CO2, -0.5 ATP, -0.5 NADH, 0.6 NADPH, -0.5 FADH2
Mode 23 = -0.6 GLCx, OXOGx, -0.4 CO2, -HCO3, -ATP, -NADH, 1.2 NADPH, -FADH2
Mode 28 = -0.5 GLCx, OXOGx, -CO2, -HCO3, -1.5 ATP, -1.5 NADH, 0.5 NADPH, -FADH2
Mode 41 = -0.5 GLCx, OXOGx, -CO2, -HCO3, -ATP, -NADH, -FADH2
Mode 34 = -0.5 CO2, 0.5 HCO3, -0.5 ATP, -2 NADH, 2.5 NADPH, -0.5 FADH2
Mode 36 = -0.5 ATP, -2 NADH, 2.5 NADPH, -0.5 FADH2
Mode 48 = -0.5 CO2, 0.5 HCO3, 0.5 ATP, -NADH, 1.5 NADPH, -0.5 FADH2
Mode 50 = 0.5 ATP, -NADH, 1.5 NADPH, -0.5 FADH2
Mode 35 = -0.33 OXOGx, 1.17 CO2, 0.5 HCO3, 0.5 ATP, 2.5 NADPH, 0.17 FADH2
Mode 38 = -0.25 OXOGx, 0.75 CO2, 0.5 HCO3, 0.25 ATP, -0.5 NADH, 2.5 NADPH
Mode 39 = -0.25 OXOGx, CO2, 0.25 HCO3, 0.25 ATP, -0.5 NADH, 2.5 NADPH
Mode 6 = -3 GLCx, OXOGx, 13 CO2, 3 ATP, 9 NADH, 18 NADPH, FADH2
Mode 21 = -GLCx, 0.5 OXOGx, 4 CO2, -0.5 HCO3, 0.5 ATP, 2 NADH, 6 NADPH
Mode 22 = -GLCx, 0.5 OXOGx, 3.5 CO2, 0.5 ATP, 2 NADH, 6 NADPH
Mode 13 = -1.5 GLCx, OXOGx, 4.5 CO2, -0.5 HCO3, 1.5 NADH, 9 NADPH, -0.5 FADH2
Mode 40 = -0.25 GLCx, 1.5 CO2, 0.25 ATP, 0.25 NADH, 2.75 NADPH
Mode 2 = -0.5 GLCx, 0.5 OXOGx, 0.5 CO2, -0.5 ATP, -0.5 NADH, 3 NADPH, -0.5 FADH2
Mode 18 = -GLCx, OXOGx, 2 CO2, -HCO3, -ATP, -NADH, 6 NADPH, -FADH2
Mode 19 = -GLCx, 6 CO2, 2 ATP, 5 NADH, 6 NADPH, FADH2
Mode 20 = -GLCx, 5 CO2, HCO3, 2 ATP, 5 NADH, 6 NADPH, FADH2
Mode 0 = -0.25 GLCx, 1.5 CO2, 1.5 ATP, 1.5 NADH, 1.5 NADPH
Mode 54 = -0.25 GLCx 1.5 CO2, 1.25 ATP, 1.25 NADH, 1.75 NADPH
Mode 47 = -0.5 OXOGx, 2 CO2, 0.5 HCO3, 2 ATP, 2 NADH, 1.5 NADPH, 0.5 FADH2
Mode 51 = -0.5 OXOGx, 1.5 CO2, HCO3, 2 ATP, 2 NADH, 1.5 NADPH, 0.5 FADH2
Mode 33 = -0.5 OXOGx, 2 CO2, 0.5 HCO3, ATP, NADH, 2.5 NADPH, 0.5 FADH2
Mode 37 = -0.5 OXOGx, 1.5 CO2, HCO3, ATP, NADH, 2.5 NADPH, 0.5 FADH2
Mode 49 = -0.33 OXOGx, 1.17 CO2, 0.5 HCO3, 1.5 ATP, NADH, 1.5 NADPH, 0.17 FADH2
Mode 52 = -0.25 OXOGx, 0.75 CO2, 0.5 HCO3, 1.25 ATP, 0.5 NADH, 1.5 NADPH
Mode 53 = -0.25 OXOGx, CO2, 0.25 HCO3, 1.25 ATP, 0.5 NADH, 1.5 NADPH
Mode 43 = -0.5 GLCx, 2 CO2, HCO3, 2 ATP, 5 NADH, FADH2
Mode 30 = -0.5 GLCx, 2 CO2, HCO3, 1.5 ATP, 4.5 NADH, 0.5 NADPH, FADH2
Mode 25 = -0.6 GLCx, 2.6 CO2, HCO3, 2 ATP, 5 NADH, 1.2 NADPH, FADH2
Mode 42 = -0.5 GLCx, 3 CO2, 2 ATP, 5 NADH, FADH2
Mode 29 = -0.5 GLCx, 3 CO2, 1.5 ATP, 4.5 NADH, 0.5 NADPH, FADH2
Mode 24 = -0.6 GLCx, 3.6 CO2, 2 ATP, 5 NADH, 1.2 NADPH, FADH2
Mode 9 = -1.5 GLCx, OXOGx, 4 CO2, 3 ATP, 9 NADH, FADH2
Mode 8 = -1.5 GLCx, OXOGx, 4 CO2, 1.5 ATP, 7.5 NADH, 1.5 NADPH, FADH2
Mode 7 = -1.8 GLCx, OXOGx, 5.8 CO2, 3 ATP, 9 NADH, 3.6 NADPH, FADH2
Mode 17 = -OXOGx, 3 CO2, 2 HCO3, 3 ATP, 6 NADH, 2 FADH2
Mode 11 = -OXOGx, 4 CO2, HCO3, 3 ATP, 6 NADH, 2 FADH2
Mode 12 = -OXOGx, 2 CO2, 3 HCO3, 3 ATP, 6 NADH, 2 FADH2
Mode 31 = -0.5 GLCx, 0.5 OXOGx, CO2, -0.5 HCO3, 1.5 NADH, 0.5 NADPH
Mode 26 = -0.6 GLCx, 0.5 OXOGx, 1.6 CO2, -0.5 HCO3, 0.5 ATP, 2 NADH, 1.2 NADPH
Mode 27 = -0.6 GLCx, 0.5 OXOGx, 1.1 CO2, 0.5 ATP, 2 NADH, 1.2 NADPH
Mode 32 = -0.5 GLCx, 0.5 OXOGx, 0.5 CO2, 1.5 NADH, 0.5 NADPH
Mode 45 = -0.5 GLCx, 0.5 OXOGx, CO2, -0.5, HCO3, 0.5 ATP, 2 NADH
Mode 46 = -0.5 GLCx, 0.5 OXOGx, 0.5 CO2, 0.5 ATP, 2 NADH
Mode 16 = -0.75 GLCx, OXOGx, -0.5 HCO3, 1.5 NADH, -0.5 FADH2
Mode 15 = -0.75 GLCx, OXOGx, -0.5 HCO3, -0.75 ATP, 0.75 NADH, 0.75 NADPH, -0.5 FADH2
Mode 14 = -0.9 GLCx, OXOGx, 0.9 CO2, -0.5 HCO3, 1.5 NADH, 1.8 NADPH, -0.5 FADH2
1
2
3
4
5
1

Figure 3.4 – Dendrogram to show all the EMs from the minimal 2-oxoglutarate model clustered by angle according to their net external metabolite usage. Branch colouration indicates differing net carbon stoichiometries as explained in Section 3.4.1. Individual clusters have been numbered for interpretative purposes and will be referred to in the discussion section.

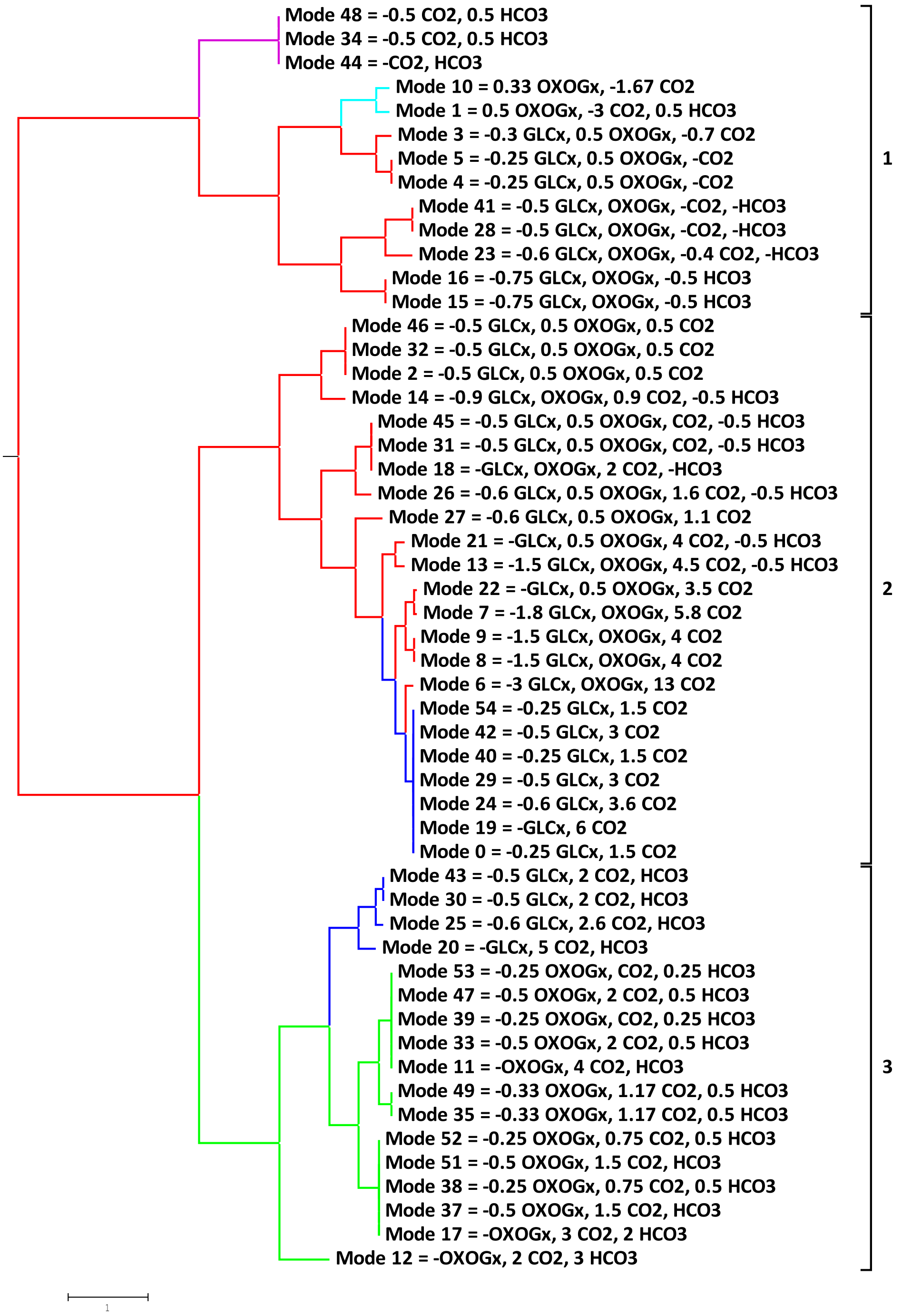

Mode 48 = -0.5 CO2, 0.5 HCO3
Mode 34 = -0.5 CO2, 0.5 HCO3
Mode 44 = -CO2, HCO3
Mode 10 = 0.33 OXOGx, -1.67 CO2
Mode 1 = 0.5 OXOGx, -3 CO2, 0.5 HCO3
Mode 3 = -0.3 GLCx, 0.5 OXOGx, -0.7 CO2
Mode 5 = -0.25 GLCx, 0.5 OXOGx, -CO2
Mode 4 = -0.25 GLCx, 0.5 OXOGx, -CO2
Mode 41 = -0.5 GLCx, OXOGx, -CO2, -HCO3
Mode 28 = -0.5 GLCx, OXOGx, -CO2, -HCO3
Mode 23 = -0.6 GLCx, OXOGx, -0.4 CO2, -HCO3
Mode 16 = -0.75 GLCx, OXOGx, -0.5 HCO3
Mode 15 = -0.75 GLCx, OXOGx, -0.5 HCO3
Mode 46 = -0.5 GLCx, 0.5 OXOGx, 0.5 CO2
Mode 32 = -0.5 GLCx, 0.5 OXOGx, 0.5 CO2
Mode 2 = -0.5 GLCx, 0.5 OXOGx, 0.5 CO2
Mode 14 = -0.9 GLCx, OXOGx, 0.9 CO2, -0.5 HCO3
Mode 45 = -0.5 GLCx, 0.5 OXOGx, CO2, -0.5 HCO3
Mode 31 = -0.5 GLCx, 0.5 OXOGx, CO2, -0.5 HCO3
Mode 18 = -GLCx, OXOGx, 2 CO2, -HCO3
Mode 26 = -0.6 GLCx, 0.5 OXOGx, 1.6 CO2, -0.5 HCO3
Mode 27 = -0.6 GLCx, 0.5 OXOGx, 1.1 CO2
Mode 21 = -GLCx, 0.5 OXOGx, 4 CO2, -0.5 HCO3
Mode 13 = -1.5 GLCx, OXOGx, 4.5 CO2, -0.5 HCO3
Mode 22 = -GLCx, 0.5 OXOGx, 3.5 CO2
Mode 7 = -1.8 GLCx, OXOGx, 5.8 CO2
Mode 9 = -1.5 GLCx, OXOGx, 4 CO2
Mode 8 = -1.5 GLCx, OXOGx, 4 CO2
Mode 6 = -3 GLCx, OXOGx, 13 CO2
Mode 54 = -0.25 GLCx, 1.5 CO2
Mode 42 = -0.5 GLCx, 3 CO2
Mode 40 = -0.25 GLCx, 1.5 CO2
Mode 29 = -0.5 GLCx, 3 CO2
Mode 24 = -0.6 GLCx, 3.6 CO2
Mode 19 = -GLCx, 6 CO2
Mode 0 = -0.25 GLCx, 1.5 CO2
Mode 43 = -0.5 GLCx, 2 CO2, HCO3
Mode 30 = -0.5 GLCx, 2 CO2, HCO3
Mode 25 = -0.6 GLCx, 2.6 CO2, HCO3
Mode 20 = -GLCx, 5 CO2, HCO3
Mode 53 = -0.25 OXOGx, CO2, 0.25 HCO3
Mode 47 = -0.5 OXOGx, 2 CO2, 0.5 HCO3
Mode 39 = -0.25 OXOGx, CO2, 0.25 HCO3
Mode 33 = -0.5 OXOGx, 2 CO2, 0.5 HCO3
Mode 11 = -OXOGx, 4 CO2, HCO3
Mode 49 = -0.33 OXOGx, 1.17 CO2, 0.5 HCO3
Mode 35 = -0.33 OXOGx, 1.17 CO2, 0.5 HCO3
Mode 52 = -0.25 OXOGx, 0.75 CO2, 0.5 HCO3
Mode 51 = -0.5 OXOGx, 1.5 CO2, HCO3
Mode 38 = -0.25 OXOGx, 0.75 CO2, 0.5 HCO3
Mode 37 = -0.5 OXOGx, 1.5 CO2, HCO3
Mode 17 = -OXOGx, 3 CO2, 2 HCO3
Mode 12 = -OXOGx, 2 CO2, 3 HCO3
1
2
3
1

Figure 3.5 – Dendrogram to show all the EMs from the minimal 2-oxoglutarate model clustered by angle according to their net external metabolite usage of carbon entities. Branch colouration indicates differing net carbon stoichiometries as explained in Section 3.4.1. Individual clusters have been numbered for interpretative purposes and will be referred to in the discussion section.

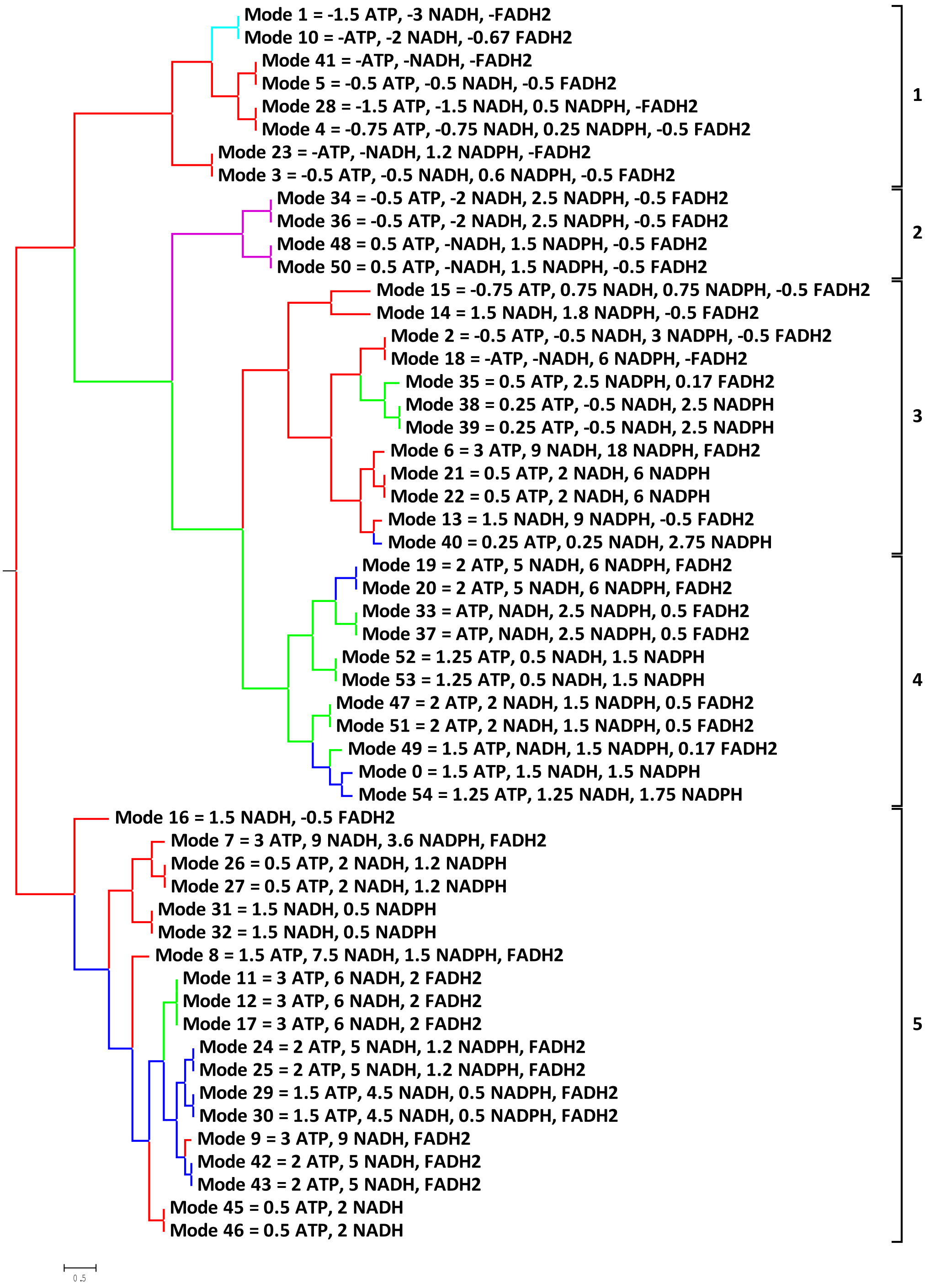

Mode 1 = -1.5 ATP, -3 NADH, -FADH2
Mode 10 = -ATP, -2 NADH, -0.67 FADH2
Mode 41 = -ATP, -NADH, -FADH2
Mode 5 = -0.5 ATP, -0.5 NADH, -0.5 FADH2
Mode 28 = -1.5 ATP, -1.5 NADH, 0.5 NADPH, -FADH2
Mode 4 = -0.75 ATP, -0.75 NADH, 0.25 NADPH, -0.5 FADH2
Mode 23 = -ATP, -NADH, 1.2 NADPH, -FADH2
Mode 3 = -0.5 ATP, -0.5 NADH, 0.6 NADPH, -0.5 FADH2
Mode 34 = -0.5 ATP, -2 NADH, 2.5 NADPH, -0.5 FADH2
Mode 36 = -0.5 ATP, -2 NADH, 2.5 NADPH, -0.5 FADH2
Mode 48 = 0.5 ATP, -NADH, 1.5 NADPH, -0.5 FADH2
Mode 50 = 0.5 ATP, -NADH, 1.5 NADPH, -0.5 FADH2
Mode 15 = -0.75 ATP, 0.75 NADH, 0.75 NADPH, -0.5 FADH2
Mode 14 = 1.5 NADH, 1.8 NADPH, -0.5 FADH2
Mode 2 = -0.5 ATP, -0.5 NADH, 3 NADPH, -0.5 FADH2
Mode 18 = -ATP, -NADH, 6 NADPH, -FADH2
Mode 35 = 0.5 ATP, 2.5 NADPH, 0.17 FADH2
Mode 38 = 0.25 ATP, -0.5 NADH, 2.5 NADPH
Mode 39 = 0.25 ATP, -0.5 NADH, 2.5 NADPH
Mode 6 = 3 ATP, 9 NADH, 18 NADPH, FADH2
Mode 21 = 0.5 ATP, 2 NADH, 6 NADPH
Mode 22 = 0.5 ATP, 2 NADH, 6 NADPH
Mode 13 = 1.5 NADH, 9 NADPH, -0.5 FADH2
Mode 40 = 0.25 ATP, 0.25 NADH, 2.75 NADPH
Mode 19 = 2 ATP, 5 NADH, 6 NADPH, FADH2
Mode 20 = 2 ATP, 5 NADH, 6 NADPH, FADH2
Mode 33 = ATP, NADH, 2.5 NADPH, 0.5 FADH2
Mode 37 = ATP, NADH, 2.5 NADPH, 0.5 FADH2
Mode 52 = 1.25 ATP, 0.5 NADH, 1.5 NADPH
Mode 53 = 1.25 ATP, 0.5 NADH, 1.5 NADPH
Mode 47 = 2 ATP, 2 NADH, 1.5 NADPH, 0.5 FADH2
Mode 51 = 2 ATP, 2 NADH, 1.5 NADPH, 0.5 FADH2
Mode 49 = 1.5 ATP, NADH, 1.5 NADPH, 0.17 FADH2
Mode 0 = 1.5 ATP, 1.5 NADH, 1.5 NADPH
Mode 54 = 1.25 ATP, 1.25 NADH, 1.75 NADPH
Mode 16 = 1.5 NADH, -0.5 FADH2
Mode 7 = 3 ATP, 9 NADH, 3.6 NADPH, FADH2
Mode 26 = 0.5 ATP, 2 NADH, 1.2 NADPH
Mode 27 = 0.5 ATP, 2 NADH, 1.2 NADPH
Mode 31 = 1.5 NADH, 0.5 NADPH
Mode 32 = 1.5 NADH, 0.5 NADPH
Mode 8 = 1.5 ATP, 7.5 NADH, 1.5 NADPH, FADH2
Mode 11 = 3 ATP, 6 NADH, 2 FADH2
Mode 12 = 3 ATP, 6 NADH, 2 FADH2
Mode 17 = 3 ATP, 6 NADH, 2 FADH2
Mode 24 = 2 ATP, 5 NADH, 1.2 NADPH, FADH2
Mode 25 = 2 ATP, 5 NADH, 1.2 NADPH, FADH2
Mode 29 = 1.5 ATP, 4.5 NADH, 0.5 NADPH, FADH2
Mode 30 = 1.5 ATP, 4.5 NADH, 0.5 NADPH, FADH2
Mode 9 = 3 ATP, 9 NADH, FADH2
Mode 42 = 2 ATP, 5 NADH, FADH2
Mode 43 = 2 ATP, 5 NADH, FADH2
Mode 45 = 0.5 ATP, 2 NADH
Mode 46 = 0.5 ATP, 2 NADH
1
2
3
4
5
0.5

Figure 3.6 – Dendrogram to show all the EMs from the minimal 2-oxoglutarate model clustered by angle according to their net external metabolite usage of energy and reducing equivalents. Branch colouration indicates differing net carbon stoichiometries as explained in Section 3.4.1. Individual clusters have been numbered for interpretative purposes and will be referred to in the discussion section.

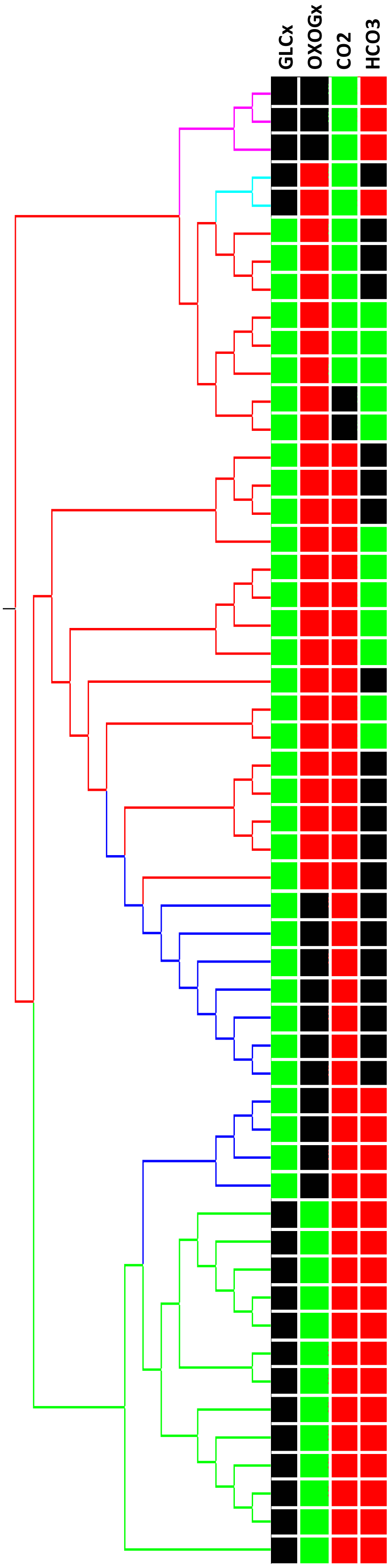


Figure 3.7 – Dendrogram and matrix visualisation for the EMs from the minimal 2-oxoglutarate model as generated in Figure 3.5, however, branch information has been replaced with the corresponding EM stoichiometry matrix. Matrix has been coloured according to coefficient value i.e. =0 (black), <0 (green) and >0 (red).

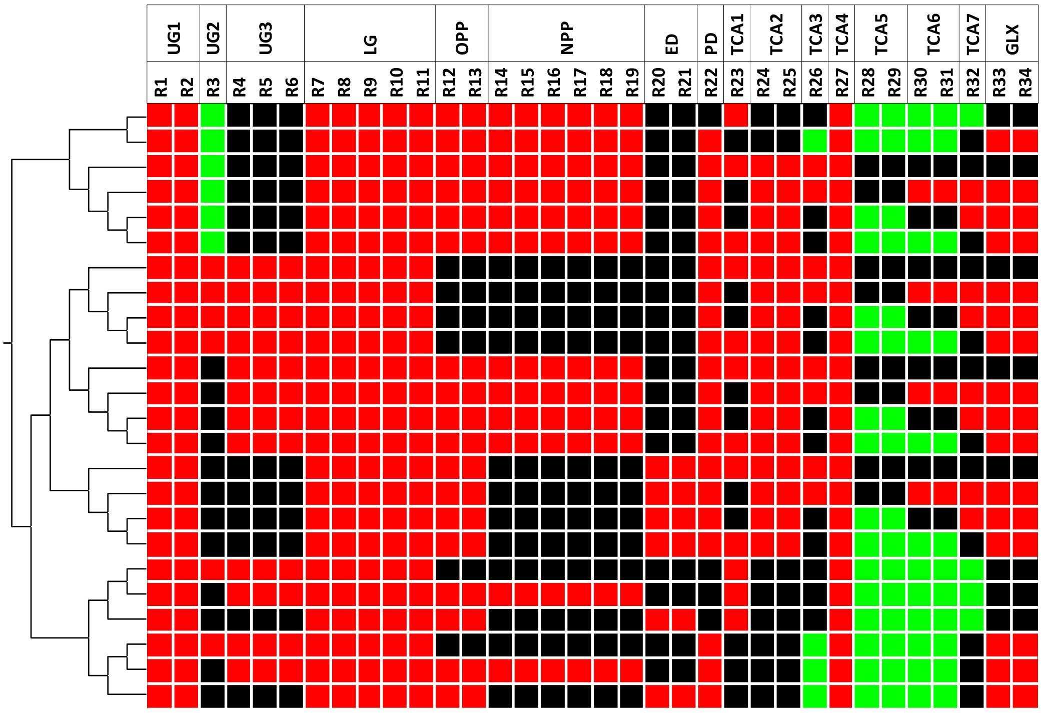


Figure 3.8 – Dendrogram and matrix visualisation to show the reaction usage of the 24 modes that produce 2-oxoglutarate from glucose. Modes were clustered by angle according to their reaction usage. Reactions have been boxed to indicate enzyme subsets (Table 3.1). Matrix coefficients have been coloured according to value i.e. =0 (black), <0 (green) and >0 (red).

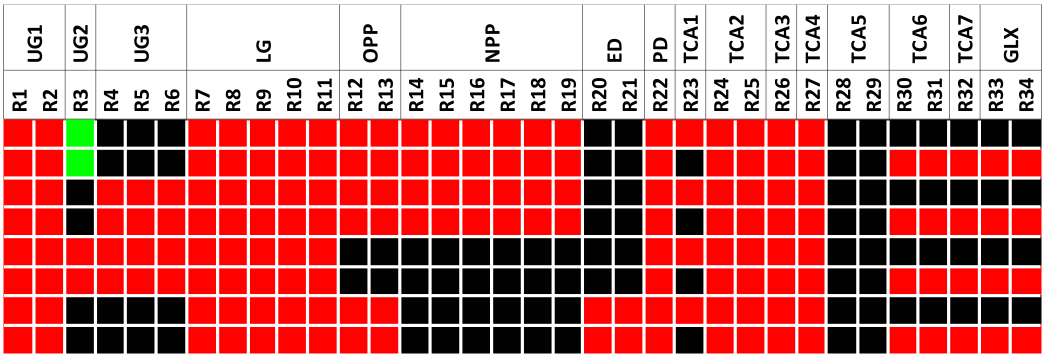


Figure 3.9 – Matrix visualisation to show the reaction usage of the 8 modes that produce 2-oxoglutarate from glucose without carbon dioxide fixation. Reactions have been boxed to indicate enzyme subsets (Table 3.1). Matrix coefficients have been coloured according to value i.e. =0 (black), <0 (green) and >0 (red).

analysis was that enzyme subsets may be functional units for performing coordinated metabolic regulation by sharing some common regulation at the level of the genome. Therefore, the results from the enzyme subsets analysis were encouraging not only for the study in question but also as an indicator of their potential association with gene expression studies.

The remainder of this section will focus on the application of the described clustering techniques for the biological interpretation of the EMs dataset from the minimal model. For the dendrogram clustered according to reaction usage in Figure 3.3, a number of cluster-specific inferences can be made:

- **Cluster 1**
  The shortest mode (Mode 44) was found in this cluster and it was found to use pyruvate dehydrogenase (R_{22}), reverse TCA and glyoxylate cycle reactions to produce bicarbonate using carbon dioxide with no net production of energy or reducing equivalents. The remainder of modes in this cluster all utilised glucose-6-phosphate isomerase (R_3) in a negative direction and the entire pentose phosphate pathway. Those that included glucose consumption also included lower glycolytic reactions to convert the glyceraldehyde-3 phosphate generated from the pentose phosphate pathway into pyruvate.

- **Cluster 2**
  All the modes in this cluster use TCA cycle reactions in the backward direction to produce 2-oxoglutarate. The shortest mode identifies a route that completes a reverse turn of the TCA cycle for the production of 2-oxoglutarate from carbon dioxide and a net consumption of ATP, NADH and $FADH_2$.

- **Cluster 3**
  All of the modes in this cluster use pyruvate dehydrogenase (R_{22}), citrate synthase (R_{24}) and aconitase (R_{25}) with the majority producing 2-oxoglutarate from glucose. The differing net stoichiometries can be attributed to the pathway variations used to generate pyruvate for the pyruvate dehydrogenase reaction. As elaborated later in this section, this cluster consists of the majority of modes that produce 2-oxoglutarate from glucose without carbon dioxide fixation (Figure 3.9).

- **Cluster 4**
  All of the modes in this cluster use 3 upper glycolytic reactions (R_3–R_5), lower glycolysis, and the non-oxidative pentose phosphate pathway in conjunction with the reverse Entner-Doudoroff pathway. Reverse usage of TCA cycle

reactions is also noted for most modes in order to completely oxidise 2-oxoglutarate to carbon dioxide and bicarbonate.

From the list above it has been demonstrated that clustering by reaction profile identifies patterns of similar pathway usage that reflect the stoichiometric constraints imposed at steady state. Furthermore, clustering by net stoichiometry is another means with which to group modes to investigate metabolic aspects such as metabolite yield. Figure 3.4 shows the EMs dataset from the minimal model clustered according to net carbon, energy and reducing equivalents. Most of the modes in the first cluster produce 2-oxoglutarate from carbon derived from either glucose, carbon dioxide and/or bicarbonate and consume ATP, NADH and $FADH_2$ in the process. Although the energy and reducing value of these modes may not be of interest, the fact that they form a distinct cluster is encouraging whereby user discretion decides whether they are investigated further or not. On the other hand, modes with the highest yield of 2-oxoglutarate, and energy and reducing equivalents were found in the fifth cluster. Further refinement of the criteria used to generate the dendrogram in Figure 3.4 can be used to cluster modes according to specific moieties of interest. Consequently, a clustering was carried out without the energy and reducing equivalents in the original EMs stoichiometry matrix. From the colouration in the resulting dendrogram (Figure 3.5) it follows that restricting the clustering criteria based on the metabolic properties of interest provides an improved classification according to the yield of just carbon entities. Additionally, the coloured matrix counterpart of Figure 3.5 (Figure 3.7) can be used to highlight the usefulness of a descriptive visualisation as opposed to textual information being conveyed on the branches of the dendrogram. The patterns in the visualisation immediately indicate which category of modes are present in a cluster, and which modes are using which metabolites.

Excluding the glucose transport reaction (R_1) there were four other reactions that were found to be common to all those modes utilising glucose to produce 2-oxoglutarate (Figure 3.8). These four glycolysis reactions were found in the same enzyme subset catalysing the net conversion of glyceraldehyde-3-phosphate to pyruvate (R_6–R_{10}). In Figure 3.8, a block type colouration is immediately visible for modes in different clusters. With the use of this representation it is much easier to understand not only how the modes have been clustered but also to visualise the potential pathway variations that can occur within each cluster. On the right hand side of the matrix a recurring ladder-like pattern is immediately noticeable for a group of 16 modes. Regardless of the pathway used to obtain pyruvate, there can be four possible variations in terms of the TCA and glyoxylate cycles to produce 2-oxoglutarate from glucose, including:

1. directly from the first few reactions in the TCA cycle (R_{23}–R_{27}).

2. without pyruvate carboxylase (R_{23}) but forward usage of most TCA cycle (R_{24}–R_{27} and R_{30}–R_{32}) and glyoxylate cycle reactions to recycle oxaloacetate.

3. without pyruvate carboxylase (R_{23}) but usage of TCA cycle reactions in the forward (R_{24}, R_{25}, R_{27} and R_{32}) and backward (R_{28} and R_{29}) directions, and the glyoxylate cycle to recycle oxaloacetate.

4. with the usage of TCA cycle reactions in the forward (R_{23}–R_{25} and R_{27}) and backward (R_{28} and R_{31}) directions, and the glyoxylate cycle to recycle oxaloacetate.

Most of the modes that produce 2-oxoglutarate from glucose involved the fixation of external carbon dioxide. Interestingly, only 8 modes were found to produce 2-oxoglutarate by utilising carbon derived exclusively from glucose. Using the coloured matrix representation of these modes in Figure 3.9, a number of inferences can be made. Firstly, there are 10 reactions common to all the modes. They include the glucose import (R_1) and 2-oxoglutarate export (R_{27}) reactions along with lower glycolysis (R_7–R_{11}), pyruvate dehydrogenase (R_{22}) and the top half of the TCA cycle (R_{24} and R_{25}). Secondly, there are two modes which use the phosphoglucose isomerase catalysed reaction (R_3) in a backwards direction. These two modes identify the complete oxidation of glucose to 2-oxoglutarate using the typical pentose phosphate and glycolytic pathway reactions, with or without the glyoxylate cycle. The mode involving the glyoxylate cycle (Mode 6) was found to have the highest yield within the entire EM dataset for the production of energy (3 ATP) and reducing equivalents (9 NADH, 18 NADPH and $FADH_2$). However, the carbon utilisation indicated that 3 molecules of glucose were required per 2-oxoglutarate formed with the remaining carbon given off as carbon dioxide. Thirdly, on further inspection, all the typical routes from glucose consumption to 2-oxoglutarate production via central metabolism can be identified (i.e. odd numbered rows) and include:

- pentose phosphate pathway, lower glycolysis, pyruvate dehydrogenase and the first few reactions of the TCA cycle.

- upper and lower glycolysis, pyruvate dehydrogenase and the first few reactions of the TCA cycle.

- upper and lower glycolysis, pentose phosphate pathway, pyruvate dehydrogenase and the first few reactions of the TCA cycle.

- oxidative pentose phosphate pathway, Entner-Duordoff pathway, lower glycolysis, pyruvate dehydrogenase and the first few reactions of the TCA cycle.

Fourthly, there is a notable repeat in the colouring of Figure 3.9 which can be attributed to the pathways listed above, in alternation with their glyoxylate cycle counterparts. To generate oxaloacetate, steady state constraints restrict the glyoxylate cycle modes to bypass pyruvate carboxylase (R_{22}) and instead use the glyoxylate cycle in conjunction with TCA cycle reactions (R_{28}–R_{30}). Overall the modes for the production of 2-oxoglutarate from glucose seem to reflect those that would be viable in *S. erythraea*. The combination of pathways that are active within a mode at any one time highlight expected biochemical possibilities. To keep with the objectives of building the minimal model the next step will be to confirm all of the reported findings with the genome-scale model of *S. erythraea*.

CHAPTER 4

Reconstruction of Genome-Scale Metabolic Networks

4.1 Introduction

In recent years there has been a shift in biology from a component-based perspective to a systems view of the cell [104]. Consequently, the amount of information available on metabolic pathways for different organisms is increasing very rapidly due to the unprecedented increase in the number of sequenced genomes - many of which are microbial. However, it has become increasingly clear that knowing the complete set of genes that code for an organism is not sufficient for understanding how it is biochemically organised. One approach to understanding the molecular physiology of an organism is to reconstruct genome-scale *in silico* models of metabolism based on its available genome annotation and biochemical literature [18]. The initial reconstruction is obtained through sequence similarity searching for all the genes that code for metabolic enzymes and, subsequently identifying the reactions that all of these enzymes catalyse [105] (Section 4.2). To this end, the low data requirements for structural modelling techniques and the relative sparsity of kinetic information makes them especially attractive for the investigation of genome-scale models.

Constraint-based metabolic models have been reconstructed for several well-studied organisms (reviewed in [19]) including *E. coli* [20], *H. influenzae* [74], *H. pylori* [106], and *S. cerevisiae* [73]. Automated methods for the genome-scale reconstruction of metabolic models (Section 4.3) are only sufficient for generating draft-quality networks with some missing functionality due to gaps in the genome annotation (Section 4.4.2). Moreover, the primary extraction of biochemical information from online repositories (Section 4.2) is by no means standardised or biochemically consistent, ultimately, leading to further complications and the need for extensive manual intervention (Section 4.5.2). On the whole, in terms of accuracy, analyses carried out on rough models can only produce rough predictions. This chapter aims to address the process of building genome-scale structural models, the problems encountered therein and potential steps that can be taken to improve

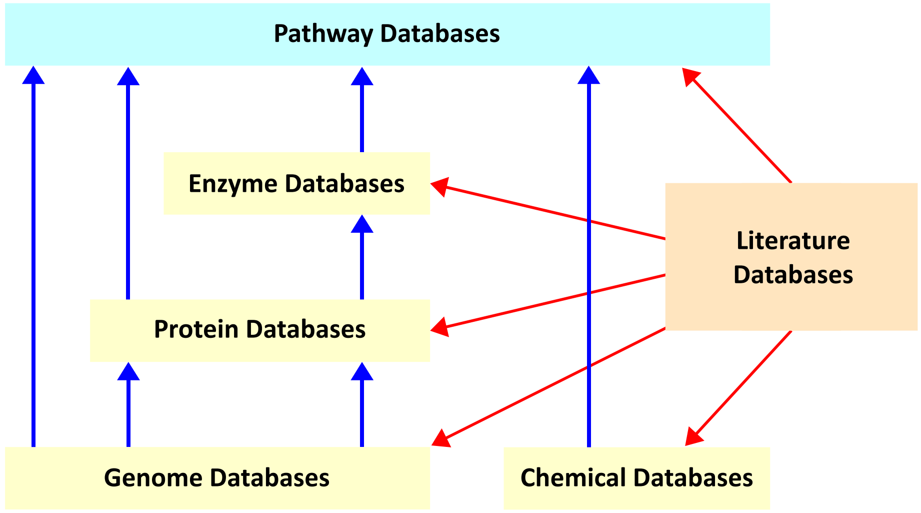


Figure 4.1 – Diagram to show the linkage of information between different types of biological databases. Databases that contain data about specific biochemical entities have been highlighted in yellow, and arrows indicate the primary flow of information between databases.

the quality of the model in order to generate more reliable phenotypic predictions.

4.2 Online databases

The combination of reductionist as well as holistic experimentation has fuelled the development of biological databases for the storage, curation and retrieval of biochemical information. Furthermore, the revolution in computer technology and memory storage capability along with the paralleled development of various high-throughput data collection technologies has allowed for the simultaneous investigation of individual compounds, enzymes, reactions and their collective representation in the form of pathways. The classification of biological databases is often based on their biochemical content, however, extensive links are usually provided to other databases for alternative or supplementary information.

Whereas enzyme databases archive information about enzymes and their properties, pathway databases contain a substantial amount of reaction data that is tightly integrated with organism-specific genomic and proteomic information (Figure 4.1) (see Pathway Resource List†). Several excellent pathway databases are available for the reconstruction of metabolic networks, whether small hand-made models or genome-scale. There are two fundamental types of metabolic database: general-purpose (e.g. KEGG† [107] and MetaCyc† [9]) and organism-specific (e.g. EcoCyc† [11] and *Saccharomyces* Genome Database† [12]). Wittig and De Beuckelaer proposed an analysis and comparison of existing metabolic pathway databases [108].

4.2.1 The KEGG database

KEGG represents the most comprehensive publicly-available bioinformatics resource of combined information on genes, proteins, metabolites, reactions and pathways [8]. This information is divided into three main databases:

1. *GENES*
   Primarily derived from publicly available resources (i.e. mostly NCBI RefSeq [109]) and contains organism-related gene information and their functional assignment for all complete genomes and some partial genomes.

2. *LIGAND*
   Contains information about compounds, reactions and enzymes [110]. It is composed of 6 other databases depending on the type of biochemical entity. These are the COMPOUND, DRUG, GLYCAN, REACTION, RPAIR, and ENZYME databases. Naming conventions and chemical structure storage are based on standard formats and, as with any other entry in KEGG, individual entries have their own unique identifier.

3. *PATHWAY*
   A collection of manually drawn pathway maps to aid in the visualisation of the enzymes and reactions participating in particular sections of metabolism (e.g. glycolysis) [107]. Enzymes or reactions of interest can be superimposed on the reference pathways already provided, which is particularly useful for viewing organism-specific pathways of interest.

Where possible all entries in KEGG are interconnected to its other internal databases. For example, the entry for water (KEGG identifier: C00001) also has links to the ENZYME, REACTION and PATHWAY databases. Furthermore, links to external databases are integrated within the DBGET [111] integrated database retrieval system available on the KEGG website. With regard to organism-specific data there is a KEGG ORGANISMS section, which is further divided into eukaryotic and prokaryotic subsections. The entire dataset stored in the KEGG database or information regarding individual organisms can be downloaded in flat file format from the KEGG GENES ftp site[†]. As highlighted in the next section, this makes it easier for the initial reconstruction of genome-scale metabolic models since the association between genes, enzymes and reactions for a particular organism are already defined.

4.2.2 The BioCyc family of databases

BioCyc contains a collection of organism-specific databases and a general purpose database called MetaCyc which forms a reference of enzyme, reaction and pathway descriptions for over 1500 different organisms [9]. The databases are divided into three tiers based on the quality of their annotation and ongoing curation efforts. For example, tier 1 contains the EcoCyc database which is a highly detailed metabolic pathway database that describes the genome and biochemistry of *E. coli*. EcoCyc is the most advanced publicly available organism-specific database and is the result of more than 20 person-years of effort to archive information regarding various aspects of its biochemistry [11]. The organism-specific databases in tiers 2 and 3 were generated automatically by the Pathologic program [112], with and without subsequent manual review and literature curation, respectively. Complete organism-specific metabolic maps can be visualised separately with the ability to select and view particular pathways of interest or to obtain a metabolic overview. Extensive links are provided between internal database entries and to external databases for additional literature and biochemical information.

The KEGG database is considered to be more popular with regard to its user-friendly access and the quantity of data incorporated. However, the BioCyc suite of databases, especially within the first two tiers, have a stringent curation policy with direct links to experimental evidence in the literature. BioCyc pathway visualisations are arguably more interpretable since unique graphics are provided for each possible pathway. On the other hand, KEGG only allows subsections of pathways to be highlighted on reference visualisations which are already highly intricate and lack an interpretable structure. On the whole, there are always problems, errors and inconsistencies within one single database and between different databases. A more detailed analysis of the differences between KEGG and BioCyc will be reported in Section 4.5 with relevance to building genome-scale metabolic models.

4.3 Model reconstruction and tools

The reconstruction of genome-scale metabolic models can take advantage of well-populated biochemical databases; they can be queried for information retrieval and analysed with the use of computer programming languages. The initial steps entail the identification of a procedure with which to reconstruct the model of interest. Traditional model reconstruction techniques focus on a ‘bottom-up’ concept whereby literature-derived biochemical knowledge is individually combined to establish larger systems-level models [104]. With the availability of genome-scale data, modern methods permit 'top-down' approaches that begin with observations at the network-level with the ability to subsequently refine and correct the initial rough representations.

4.3.1 Reconstruction process

Given the genome sequence for an organism of interest and access to the necessary databases and analytical tools, the general procedure for obtaining a list of all enzyme catalysed reactions is as follows [113] (Figure 4.2):

1. Identification of the open reading frames (ORFs) or coding regions on a genome sequence is carried out by sequence similarity alignments. In theory, each ORF is aligned and compared with other sequences in the database to identify genes in related species of known function. Popular algorithms such as BLAST[†] [114] and FASTA[†] [115] can be used for these purposes and provide a list of 'best hits' according to the level of similarity between the aligned sequences.

2. Assigning EC numbers to the genes encoding enzymes that are likely to be expressed in the organism by querying databases such as ExPASy[†] [116].

3. Correlating the list of EC numbers obtained in the previous step with their associated reactions. General purpose pathway databases can be consulted for these purposes since they encode the relationships between EC numbers and their corresponding reactions.

The KEGG database directly provides information at all levels within this hierarchy. For example, the ftp download site for the KEGG GENES section archives organism-specific folders with flat files containing various levels of biochemical data such as a list of EC numbers. Alternatively, there are a variety of software programs and analytical tools that have been developed to improve the quality of the data yielded at individual steps or for the entire process described above (Section 4.4.2).

4.3.2 Enzyme profile databases

The most popular method for the functional annotation of genomes is based on the homology assessment with data in primary sequence databases. However, there are a number of issues with these approaches since it is difficult to determine the ancestry (i.e. may have arisen due to gene duplication events as opposed to speciation) and substrate specificity of individual enzymes between species [117]. A number of databases and associated methodologies have been developed to include

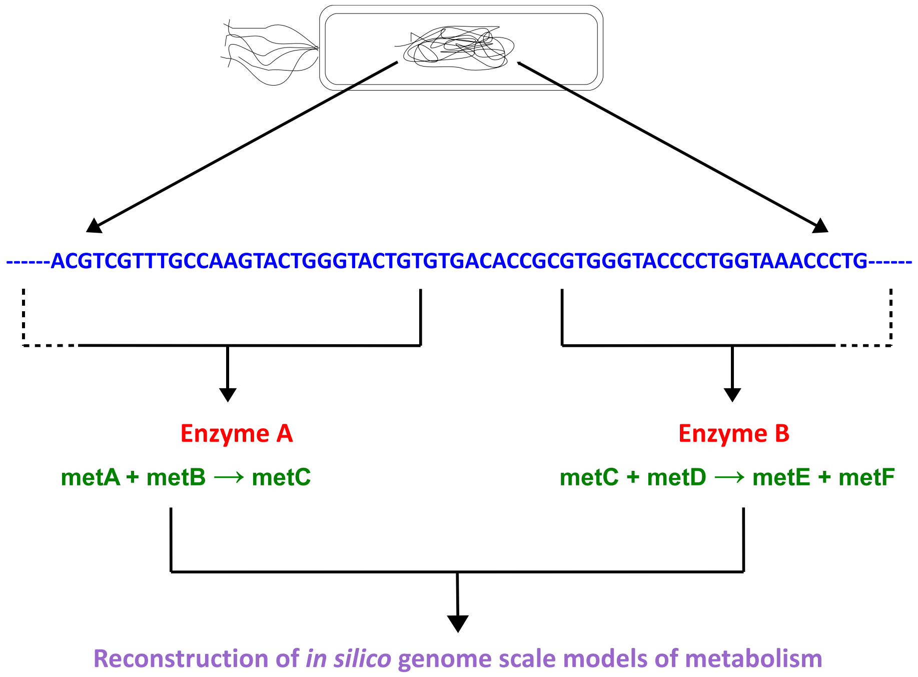

------ACGTCGTTTGCCAAGTACTGGGTACTGTGTGACACCGCGTGGGTACCCCTGGTAAACCCTG------
Enzyme A
metA + metB ⟶ metC
Enzyme B
metC + metD ⟶ metE + metF
Reconstruction of in silico genome scale models of metabolism

Figure 4.2 – Diagram to show how the genomic sequence information from a given organism can be used to reconstruct its corresponding metabolic network.

enzyme-specific profiles which contain additional discriminatory information such as lists of polypeptides involved in certain enzyme activities e.g. PRIAM† [117], metaSHARK† [118], PROSITE† database [119, 120] and InterPro† [121]. In particular, PRIAM is a method for automated enzyme detection in fully sequenced genomes, based on the classification of enzymes in the ENZYME† database [122]. To summarise, the PRIAM methodology can be broken down into five primary stages [117]. Firstly, the ENZYME databases is used to extract enzyme-specific sequence collections that are all protein sequences which share the same catalytic activity and thus, EC number. Secondly, the MKDOM program† [123] is used to identify the longest homologous segments shared within each enzyme collection. Thirdly, enzyme-specific rules are defined that determine which module(s) are required in order to infer the presence of a given enzyme. Fourthly, each of these modules is represented using a position-specific scoring matrix or 'profile' generated using PSI-BLAST† [124]. Finally, all the enzyme-specific profiles generated can then be used to search for the presence of enzymes in a genome. This step involves a homology search between each protein and the PRIAM profiles using RPS-BLAST [125], and further processing stages are carried out to generate a list of predicted enzymes.

The PRIAM methodology was compared to manual curation efforts for the functional annotation of the *Sinorhizobium meliloti* genome [117]. Out of 6204 proteins, 1460 were predicted enzyme activity, and 13 of those were found to be bifunctional multienzymes. These 1460 predicted proteins corresponded to 660

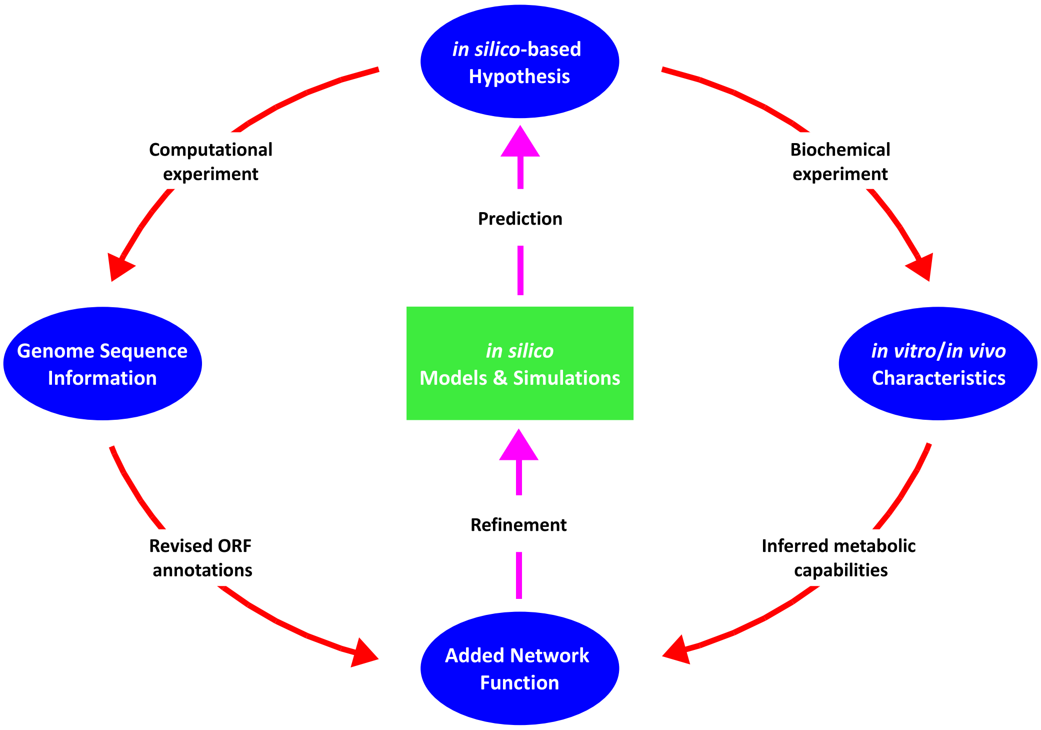

in silico-based Hypothesis
Computational experiment
Biochemical experiment
Prediction
Genome Sequence Information
in silico Models & Simulations
in vitro/*in vivo* Characteristics
Refinement
Revised ORF annotations
Inferred metabolic capabilities
Added Network Function

Figure 4.3 – Illustration to show the iterative process from the generation of a hypothesis for a model, its *in silico* translation, and the refinements required to improve its analytical usefulness. Adapted from [47].

different enzyme activities, emphasising the paralogy in the *S. meliloti* genome. PRIAM only missed 39 enzyme activities out of the 532 identified by manual annotation and it predicted an additional 167 most of which were precise representations of incomplete EC numbers (e.g. 1.1.-.-). Therefore, the use of profile-based methods such as PRIAM is advantageous to automatically recover a more precise and abundant set of enzyme activities from complete genomes. The availability of increased number of sequenced genomes will not only help to increase the specificity of the existing dataset of profiles but also to help identify new enzyme activities that were previously unrecoverable.

4.4 Modelling strategies

Once the aims for building a model have been defined, the next step is to decide on the process with which it is going to be reconstructed. To reflect the properties of the system as accurately as possible a combination of automatic and manual procedures have to be used to produce genome-scale reconstructions. Initial automated reconstructions are susceptible to the quality of the annotation, many of which are incomplete or based on other highly curated organisms. As described in the following sections, an iterative refinement [18] of the model must be carried out through a series of steps to improve the quality of the model (Figure 4.3).

4.4.1 Model definition

As mentioned earlier, a preliminary list of metabolic reactions for a given organism can be obtained from the annotated genome sequence and available biochemical information. For mathematical modelling purposes, the reaction information has to be translated into a format that is suitable for its storage, querying and analysis (e.g. '*.spy*' format) (Section 1.7.1). The necessary transport reactions have to be defined for the exchange of internal and external metabolites between intra- or intercellular compartments, respectively, and are usually defined based on the physiological properties of the organism in question. The list of transporters obtained at this stage is by no means complete and depends on the model verification procedure introduced in the next section. Incorrect or incomplete assignment of transport reactions can lead to gaps in the network which propagate into the analytical process. The presence of a small number of cellular compartments makes this task significantly easier for prokaryotes than eukaryotes.

4.4.2 Model interrogation

It is now a common notion that high-throughput methods sacrifice specificity for scale. Although initial metabolic models will have some predictive capabilities, a systematic verification has to be carried out in order to identify errors or inconsistencies within the annotation and/or individual database entries [105]. With the aim of reconstructing a model that can accurately represent *in vivo* characteristics, the validation procedure is possibly the most difficult and time-consuming task [105, 126]. To a certain extent the amount of manual effort required depends on the integrity of the initial network, which in turn depends upon the quality of the annotation. Even so, all automatically created metabolic networks are inherently incomplete and reactions are included based on three primary criteria:

1. The majority are derived from annotated genomic evidence available in pathway databases and are usually included in traditional well-characterised biochemical pathways. Although, it would be expected that reactions included from these sources are satisfactory for modelling purposes, this is not the case. As explained in Section 4.5.2.1, problems may arise through simple database errors such as unbalanced reactions which once combined to form a model can have the potential to give false confidence in the subsequent structural analyses. Other potential database inconsistencies include reaction reversibility and substrate specificity.

2. Those obtained from observations found in the literature, with the aim of filling in those metabolic properties not obtained via the previous step [105].

For example, if tyrosine is a non-essential amino acid for an organism then any missing reactions (sometimes referred to as 'gaps') in the initial network should be included to obtain a complete tyrosine-synthesis pathway. At this stage, the biochemical capabilities of the organism are also assessed with a view to defining all the transport mechanisms (i.e. for membrane diffusion, pore diffusion or active transport of metabolites across the membrane or between intracellular compartments); most of which do not have an EC number and have to be defined manually.

3. Those supported by the metabolic demands of the reconstructed network and that will require further experimental and/or genomic evidence. Reactions of this type may not be participants in traditional biochemical pathways but may be used to connect different pathways and for the use of different carbon sources.

For less-studied organisms the genome plays a more significant role in network reconstruction, and many of the enzymes are assigned based on sequence homology and await biochemical characterisation. With the added complication of roughly a quarter of all genes being species-specific, with, as yet, no known homologs, it is inevitable that a substantial fraction of genes present in the genotype have unknown function [127]. The identification of network gaps can be carried out by assessing the coupling between experimental observations and the model's ability to simulate growth on specific biomass components (tyrosine example given in step 2, above) whilst other methods exploit the availability of highly curated metabolic reconstructions to infer gene-reaction relationships in less characterised organisms [128]. A number of computational tools are now available which use a variety of techniques to help identify and fill the gaps created in automatically created metabolic networks including metaShark† [118], AUTOGRAPH [128], SEED† [129] and Pathway Tools [112] (reviewed in [130]).

One of the simplest methods for gap identification involves loading the initial model into a stoichiometry matrix and examining the connectivity within the model. Identification of network anomalies such as orphan or dead-end metabolites (Section 1.6.1.2) can be used to bridge the connectivity within the model by amending reversibility criteria (i.e. for reactions associated with dead-end metabolites) or by including additional reactions (Section 4.5.2.2). For structural analytical purposes (Section 4.4.3), the reactions within a genome-scale reconstruction should be stoichiometrically correct and essential pathways should be as complete as possible [19]. Therefore, verification of the initial metabolic reconstruction is a very important process that involves manual curation, literature and/or experimental confirmation and analytical feedback in an iterative manner.

4.4.3 Model analysis

After the development and evaluation of a validated metabolic model in the context of the available literature and biochemical information [105], it needs to be described mathematically. A model in this form can then benefit from computational tools for the analysis and integration of the data at the systems-level. The analytical phase of the modelling process is very important for updating the initial versions of the model. Refinement of the stoichiometric model in this manner can then help to formulate alternate hypotheses in the next iterative cycle - at each turn increasing its accuracy and reliability. Once in an acceptable form, various structural analytical procedures (reviewed in [19]) may be used to explore the properties of the network and its ability to reproduce and predict physiological behaviour derived from *in vivo* experimental data [131].

Structural analytical procedures on genome-scale models may be carried out from a graph-theoretical point of view to study the connectivity and global characteristics of the metabolic network [5, 31, 58] (Section 1.6.1.1). Alternatively, pathway analysis methods such as enzyme subsets analysis (Section 1.6.2.2) have been applied to the entire *E. coli* metabolic network [43]. In contrast, other pathway-based techniques including elementary modes analysis [58, 60, 61] and extreme pathways analysis [64, 132] can only be applied to organism-specific sub-networks (e.g. sugar metabolism) due to the combinatorial complexity of larger genome-scale models [40] (Section 1.6.3). Pathway analysis techniques permit the measurement of the inherent redundancy within metabolic networks which in turn is important for defining the degree of network robustness; with an end to improving metabolite yields, drug targeting or to investigate the effect of gene knockouts.

Specification of additional optimisation criteria for the genome-scale model has enabled flux balance studies (Section 1.6.3.4) to quantitatively simulate metabolic capabilities such as maximum growth rate [69, 73, 74] and the effect of genetic manipulations [20, 68, 69, 76] for various organisms. Any discrepancies between predicted genotypes and experimental phenotypes for genetic perturbations can be used to evaluate and improve the gene annotation [126]. For example, false negatives or scenarios where there is experimental growth which is not the case *in silico* may suggest missing reactions possibly catalysed by additional isozymes [133]. On the contrary, false positives can be used to implicate reactions that have been incorrectly included in the metabolic model [133].

Without the need for kinetic measurements both pathway- and optimisation-based analyses coupled with experimental confirmation can yield important insights into the completeness of the metabolic model and, ultimately, to guide metabolic engineering studies. Once again, it is necessary to emphasise the need for an extensive validation process before the outcomes of the former statement may be accomplished, especially for automatically generated organism-specific models. The following sections will highlight a recent group effort to address those issues that pose significant problems whilst generating genome-scale metabolic models from well known databases, their potential solutions and future incentives to automate the model reconstruction process.

4.5 Automated reconstruction of metabolic networks: problems

In the course of this project a publication [134] (paper included in Appendix F) was written detailing the characterisation and documentation of those types of error and inconsistencies that pose problems when building structural models from the two most popular pathway databases, KEGG and BioCyc. It was authored by M.G. Poolman, and the database analyses were collectively carried out by B.K. Bonde and A. Gevorgyan and myself, under the guidance of M.G. Poolman and D.A. Fell. With the personal insights having been gained by performing the BioCyc analyses, the results obtained in the publication will be categorised and, subsequently, discussed further. For this purpose, the following sections will highlight a number of challenges faced when carrying out genome-scale structural modelling with particular reference to the database problems/errors that have to be initially overcome.

4.5.1 Introduction and methods

Using pathway databases, the complement of enzymes present in a given organism can be accessed for building genome-scale metabolic models. While such databases serve as a primary resource for building metabolic models, the errors within them pose a new challenge. Firstly, the quality of this data is relatively poor since the pace at which genomes are being sequenced is a lot faster than the studies underway to annotate them. Secondly, each database often contains heterogeneous, incomplete, or inconsistent data that differ widely in form and content. Thirdly, the multiplicity of information sources can be overwhelming for researchers who simply wish to find information about genes or pathways of interest in a standardised fashion.

The reaction lists for five well annotated prokaryotes were derived from each database (i.e. KEGG and BioCyc) to build whole organism structural models: *E. coli* K-12 (*eco*), *Mycobacterium tuberculosis* H37Rv (*mtu*), *Vibrio cholerae* N16961 (*vch*), *H. pylori* 26695 (*hpy*) and *Bacillus anthracis* Ames (*ban*). The corresponding data files were acquired for each organism via the file download interface provided by the relevant database. Additionally, 'whole database' structural models containing all the reactions in KEGG and MetaCyc were also generated for a side-by-side comparison of the reaction set within each database. No external metabolites were defined and all reactions were regarded as irreversible. The Python programming language was used for direct database querying, automatic model generation, and graph analysis. The metabolic modelling package ScrumPy (Section 1.7.1) was used for data parsing and model interrogation. The graph connectivity analysis was based on the representation of the model's stoichiometry matrix as a bipartite graph (Section 1.6.1.1).

For the purposes of the study, the errors propagated from pathway databases to the model building process can be categorised as database-level or systems-level. The former errors are due to poor data quality (e.g. wrong reaction stoichiometry) while the latter arise while studying large metabolic systems (e.g. orphan metabolites). Based on these criteria, a classification of the specific problems reported in Table 4.1 will be provided in the next section.

4.5.2 Results and discussion

Apart from those errors that are explicitly reported in the text, all results are presented in Table 4.1. Note that all the results in the table are expressed as percentages of totals, except the entries for 'sub-graphs' which refer to the number of sub-graphs in the model. In terms of the total number of reactions and metabolites, all the models reconstructed from BioCyc were found to be considerably smaller than those generated from KEGG. For organism-specific models, the largest overall difference in content was observed for *M. tuberculosis*, whereby the BioCyc model had 494 and 747 fewer reactions and metabolites, respectively, than its KEGG counterpart. In contrast, the frequency of errors reported from BioCyc derived databases were remarkably lower when compared to those built from KEGG. However, the *E. coli* model generated from BioCyc was found to be an exception. In comparison with other organism-specific models built from BioCyc it had approximately 10 times more reactions with the same metabolite acting as a substrate and a product. Additionally, it contained 3.6 (*vch*) to 7.2 (*mtu* and *hpy*) times more connected components than the other BioCyc models. When compared to the organism-specific KEGG models the number of reactions with the same metabolite acting as a substrate and a product, and the number of connected components were still higher in the BioCyc *E. coli* model, but not as high in proportion to the other models reconstructed using BioCyc. The significance of these results will be discussed over the following sections.

Model	All	*eco*	*mtu*	*ban*	*vch*	*hpy*
KEGG						
Total reactions	6576	1532	1271	1249	1228	589
Total metabolites	5538	1691	1530	1480	1410	819
Unbalanced reactions	6.9	4.6	5.7	5.4	4.8	6.5
Same metabolite	1.5	2.3	2.6	2.7	2.6	4.1
Orphan metabolites	41.2	50.1	51.3	53.7	50.8	57.3
Dead-end metabolites	11.6	18.5	22.4	21.6	20.4	24.4
Sub-graphs	28	19	19	17	22	15
Largest sub-graph	99	98	98	98	98	96
BioCyc						
Total reactions	5071	1394	777	1007	826	536
Total metabolites	4846	1394	783	951	813	565
Unbalanced reactions	1.3	0.6	1.4	1.3	0.7	1.5
Same metabolite	2.1	16.1	1.7	1.5	1.8	1.9
Orphan metabolites	51.9	42.9	30.7	31.0	34.7	32.4
Dead-end metabolites	7.6	11.5	13.6	11.6	11.0	14.4
Sub-graphs	67	36	5	8	10	5
Largest sub-graph	98	86	99	99	99	99

Table 4.1 – Results table from [134] to show the relative amount of problems/errors encountered whilst building whole database ('All') and organism-specific models from the KEGG and BioCyc databases. The whole database analyses for BioCyc was carried out using MetaCyc. Results are expressed as percentages of totals, except 'sub-graphs' referring to the number of sub-graphs in the model. 'Same metabolite' refers to reactions involving the same metabolite as both a substrate and a product. See Section 4.5.1 for organism abbreviations.

4.5.2.1 Database-specific problems

Problems found due to either incorrect curation or annotation efforts at the level of the database can be called 'database-level' problems. Database-level problems include reactions which have an overall imbalance for atoms such as carbon, oxygen and nitrogen. For metabolic modelling purposes, these fall into two categories, those resulting from:

- *Database errors*
  Where present, the empirical formulae for all the species participating in a reaction may be used to identify the overall atomic balance. The organism-specific and whole databases for KEGG had 4–5.6% more unbalanced reactions (i.e. in proportion to the total number of reaction in each model) resulting from database errors when compared to their BioCyc counterparts. For example, KEGG reaction R05524 (EryCI) has an ammonia molecule missing in the reactants and the reaction for biotin synthase (EC:2.8.1.6) in

MetaCyc has sulphur missing from the reactants. Errors of this sort can have a direct impact on structural analysis techniques such as elementary modes. Depending on external metabolite definitions, all the resulting modes involving an unbalanced reaction may turn out to be stoichiometrically incorrect (i.e. will not obey the laws of mass conservation) and, consequently, will not reflect the true properties of the network in question.

- *Ambiguous interpretation*

  For the purposes of metabolic modelling a reaction must be defined as a biochemical conversion. Reactions that are found to have the same metabolite on both sides are incorrect at this level since a biochemical transformation has not been defined (e.g. Starch + $H_2O \longrightarrow \alpha$-D-Glucose + Starch). In the above example, starch can be seen to act as a catalyst as opposed to a participant of the reaction. Furthermore, the representation for this reaction in a stoichiometry matrix can be:

  $$\mathrm{H_2O} \longrightarrow \alpha\text{-D-Glucose}$$

  $$\mathrm{H_2O} \longrightarrow \alpha\text{-D-Glucose} + \text{Starch}$$

  $$\text{Starch} + \mathrm{H_2O} \longrightarrow \alpha\text{-D-Glucose}$$

  all of which imply a net imbalance in atomic composition. Despite instances of this type being justified as a database entry, they have an ambiguous interpretation in the metabolic modelling context. The EcoCyc database was found to have 12–14.6% more reactions with the same metabolite on both sides when compared to all the other models built in the study. The majority of these reactions had been explicitly defined as transport reactions between various cellular compartments. It is not an error to treat the same metabolite as different species in different compartments. Due to the intense curation efforts that are being undertaken with this database it is, therefore, not surprising that an extensive reaction set that encompasses the organisms ability to metabolise as well as exchange nutrients has been included.

With regard to the whole database models, the results shown in Table 4.1 indicate that the frequency of database-level errors in the MetaCyc model are lower than in the KEGG model. This may be explainable by the extensive curation efforts undertaken by the BioCyc family of databases. Database-level inconsistencies appear to be the result of a number of different classes of problem arising from the original database. The possibility exists for multiple problems originating in the database to interact with each other in the model, so that a particular problem in the model (e.g. absence of steady state flux in a given reaction) cannot be resolved until all the problems from the issuing database have been corrected.

4.5.2.2 Systems-specific problems

Even where individual reactions appear to be specified correctly, taken in isolation, problems can become magnified once the metabolic model is built. These can be classified as 'systems-level' problems. Systems-level problems reduce the maximal connectivity within the generated model and, therefore, almost certainly require extra effort from the user to improve the quality of the model.

For all the models built in this study, the percentage of orphan and dead-end metabolites were surprisingly high, peaking at 57.3% and 24.4% for the *hpy* KEGG model, respectively. The frequency of orphan metabolites for the whole database models reconstructed from MetaCyc was 51.9% and 41.2% for KEGG, highlighting that approximately half of the metabolites in each database are only associated with one reaction. This may have come about for a number of reasons. The primary factor is the lack of annotated reaction data that is required to link these metabolites with other metabolites in the corresponding metabolic network. It is an undisputed fact that the annotation for all genomes is incomplete, and it is likely that there are additional metabolic enzymes amongst the ORFs of unknown function. Once found, inclusion of these enzymes may help to reduce the number of orphan and dead-end metabolites. Secondly, database-level errors such as spelling mistakes make a normally connected metabolite an isolated entity within the network. For example, in KEGG, EC:2.4.1.36 (α,α-trehalose-phosphate synthase) was found to be associated with these reactions:

R02168
GDPglucose + D-Glucose 6-phosphate $\longleftrightarrow$ GDP + α,α'-Trehalose 6-phosphate

R06125
GDP-glucose + D-Glucose 6-phosphate $\longleftrightarrow$ Trehalose 6-phosphate + GDP

The entries for GDPglucose and GDP-glucose, and Trehalose 6-phosphate and α,α'-Trehalose 6-phosphate were indicated as being distinct, when this clearly cannot be the case. Thirdly, multiple reactions with generic and specific reactants can also have the same effect, for example (EC:1.1.1.1; alcohol dehydrogenase):

R00754
Ethanol + NAD^+ $\longleftrightarrow$ Acetaldehyde + NADH + H^+

R00623
Primary alcohol + NAD^+ $\longleftrightarrow$ Aldehyde + NADH + H^+

If the 'Primary alcohol' metabolite is only present once in the model then it becomes an orphan metabolite. Otherwise, its association with other reactions involving generic reactants will lead to the formation of a distinct connected component in the network. Errors of this type may be easy to detect if all the reactions involving a particular generic metabolite are noted, with future models being filtered for their presence. However, this process would have to be maintained between database releases (i.e. for the detection of new generic metabolites and identification of additional reactions involving those that have already been found). In analogy with those reactions that involve the same metabolite both as a product and reactant, the above example illustrates that a purely legitimate reaction at the database-level can cause problems at the systems-level.

Connected components analysis indicates the existence of more than one sub-network within a model. While it is not absolutely certain that all metabolic networks consist of one intricately connected component, owing to their heterotrophic nutrient requirements, one would expect that the *in vivo* metabolism of the organisms examined in the study are fully connected. This simply implies that the metabolites required to sustain life can all be derived from a single carbon source. The genome-scale model reconstructed using the EcoCyc database showed the worst connectivity results across both databases. This may have been due to the fact that the completeness in recording the EcoCyc enzyme complement is not always compatible with ensuring connectivity. At the systems-level this ultimately leads to a fragmented metabolism which is unlikely to represent the metabolism of a real organism. Except for the *E. coli* models, the largest sub-graph in all the other models reconstructed from either database were found to contain 96–99% of the reactions in the model. As described above, problems such as metabolites with both generic and specific names, and simple errors at the database-level have the potential to affect network connectivity whereby the inclusion of a reaction involving them may be the difference between a fully- and partially-connected system. The inclusion of metabolites with isomeric forms within structural models is also an important issue. A given compound may map to more than one KEGG compound ID, for example, glucose can be mapped to C00293 (glucose), C00031 (D-glucose), as well as C00221 (β-D-Glucose) and C00267 (α-D-Glucose). α- and β-D-glucose are different entities that are spontaneously interconvertible and are both, nonetheless, instances of D-glucose. As indicated in the publication (Appendix F), the anomeric sensitivity of enzymes towards glucose and its isomers is a quantitative rather than a qualitative effect, and, therefore, in terms of structural modelling it should be considered as the same substance. A similar example highlights the inclusion of the amino acid serine and its isomeric L- and D- forms. Additionally, errors at the level of the database could render the same metabolite with different unique identifiers (see GDP-glucose example in this section). Ultimately, without human inspection, errors of this type may lead to a reduced connectivity within the model since the reactions associated with a particular variant may be rendered redundant from the remainder of the metabolism.

4.5.3 Overall Summary

There are a number of ways with which to improve the quality of genome-scale models built from databases:

1. The most obvious solution for database-level errors is the provision of a web-based data updating service, which at present is more efficiently appointed by BioCyc than KEGG.

2. Automated local data processing is currently the method of choice for orphan and dead-end metabolite detection and removal (i.e. by defining them as external or source/sink metabolites). Other error instances (e.g. synonyms and corrected reactions) can be archived by the creation of in-house dictionaries which can subsequently be used to filter the input from database-derived models. With each database release it would be expected that the quantity of data will increase along with improvements in quality. Therefore, the filtering approach would require a periodic reviewing, for the detection of errors in the additional data, and management of those that have been corrected in the database.

3. Manual intervention is almost always required for those problems that cannot be wholly discovered and/or solved by automated means. For example, those reactions that have been automatically found to be chemically unbalanced cannot be solved in the same fashion. Unless these reactions have been tagged as unbalanced in the database or are to be excluded from the model they must be dealt with on a case-by-case basis. Orphan metabolites and dead reactions may also be reduced via manual gap filling in order to increase network connectivity.

From the results, it was found that the BioCyc group of databases exhibited fewer database-level errors. This may be attributed to the intense curation efforts that are undertaken within the BioCyc family of databases. Isolated errors which are most likely to occur at the database-level do not necessarily invalidate genome-scale models. The models built from databases such as KEGG and BioCyc are never going to be perfect but it is worth carrying out some preliminary analysis to identify those problems that may be propagated into the modelling process.

More importantly, the purpose for building these models is to investigate the properties of the system as a whole. Obvious systems-level errors such as orphan metabolites must be dealt with due to their high occurrences. At present, there are no standard methods in place to account for noisy data which, if ignored, almost certainly becomes incorporated into the ensuing structural analysis. The purpose of the study in question was not to compare the databases of choice but to provide an outline of how the data within each database affects the reconstruction of large automatically created metabolic models. As a consequence of the reported problems it has become evident that the data contained within such databases requires a considerable amount of development and modification before they can be successfully used at the systems-level.

CHAPTER 5

Phylogenetic Comparisons Based on Enzyme Complement

5.1 Introduction

Prokaryotes[1] have long been used as model organisms to understand the basic principles of life. They have also been exploited as important targets in disease treatment, biotechnology and ecology. It has been estimated that there are between 10^5–10^7 prokaryotic species on Earth [135]. Although, plant and animal species can be differentiated according to their embryological and morphological appearance, this is clearly not the case for prokaryotes - due to the relative lack of discernible morphological characters [136]. Phylogenetic analysis[2] of universally conserved DNA or protein sequences has become a powerful tool for microbial taxonomy[3]. The most popular classification method for phylogenetic reconstruction from sequence data exploits the similarity measurement of aligned homologous genes based on polymorphism information (techniques reviewed in [137]). Originating from work carried out in the late 1970's by Carl Woese and colleagues [138, 139], our current understanding of prokaryote taxonomy has stemmed primarily from the comparison of the highly conserved small subunit ribosomal RNA (rRNA) [140]. By the late 1980's, supplementary research carried out by the same group established the status of the rRNA gene as the 'ultimate molecular chronometer' [140]. Using this marker, the tripartite 'universal tree of life' was reconstructed, and was used to recognise what we now know as the three domains of life, namely Eukaryota, Eubacteria, and Archaea [141, 142]. The previous notion that prokaryotes were part of a monophyletic group was reassessed due to the split between the Archaea and Eubacteria.

[1] unicellular organisms without a membrane-bound nucleus; they include eubacterial and archaeal species.

[2] the study of evolutionary relatedness among various groups of organisms.

[3] the practice and science of classification; in the context of this study directed at prokaryotes.

5.1.1 Ribosomal RNA derived phylogenies: pros and cons

Ribosomal RNA has several properties which make it uniquely suited as a phylogenetic marker including its universal distribution, low substitution rates and ease of sequencing [140]. However, a number of problems with the use of rRNA have recently come to light. Firstly, horizontal gene transfers[4] (HGT) have been shown to be possible for this molecule [143]. Secondly, its restricted length permits mutational saturation[5]. Thirdly, rRNA phylogenies, in most cases are incapable of resolving deeper evolutionary branches [144]. In other words, the specific phylum to which an organism belongs can be readily identified which may not apply when trying to determine how different phyla are related to each other. Fourthly, the evolutionary distances rendered by comparison of a single gene are likely to differ from the phylogenetic history of the organism from which it was isolated [145]. Finally, phylogenetic trees derived from other protein-coding genes or proteins such as DNA polymerase and, transcription and translation factors (e.g. GT-Pases) [142, 146] have different topologies when compared with the corresponding 'universal' rRNA-based trees [147].

Prior to the post-genomic era, despite the recognition that comparison of different biological entities leads to different tree topologies, the validity of phylogenies based on rRNA remained undisputed. This notion was reassessed once the availability of complete genomes provided researchers with the opportunity to implement novel approaches to reconstruct phylogenetic relationships, ideally from the entire complement of genomic information.

5.1.2 Whole genome-based phylogenies

As a direct consequence of the growing number of complete genomes within all the domains of life, numerous techniques have been introduced to reconstruct whole genome-based phylogenies at the sequence-level [148]. Examples of whole genome studies involving sequence comparisons include the determination of the average similarity between orthologous[6] genes [148, 149, 150] and reconstruction of 'supertrees' based on the combination of trees derived from multiple conserved single genes [148, 151]. Rivera *et al.* [152] compared the complete set of orthologous genes between two bacteria (*E. coli* and *Synechocystis PCC6803*), a eukaryote (*S. cerevisiae*), and an archaeal species (*Methanococcus jannaschii*). They found that the *S. cerevisiae* protein synthesis genes (e.g. those responsible for translation and

[4] any process in which an organism incorporates genetic material from another organism without being the offspring of that organism.

[5] the observed number of mutations relative to the maximum amount possible.

[6] genes in different species that derive from a common ancestor whereas paralogy describes the relationship between two genes related through gene duplication.

transcription) were primarily derived from *M. jannaschii*, whereas the metabolic genes of the eukaryote (e.g. amino acid synthesis and energy metabolism) were more closely related to the two bacteria. It was becoming clearer that eukaryotes arose as a result of a chimera of ancestral eubacterial and archaeal genes [152, 153]. Studies such as this highlight that the role of vertical gene transfer[7] (VGT) may have previously been overestimated. Instead, there has been more emphasis on the importance of evolutionary processes such as HGT and lineage-specific gene loss, which at least in prokaryotes, involve most genes [147, 153, 154]. The intensive gene transfer between organisms as exemplified by HGT has led to the idea that the evolutionary history of life may actually be better represented as a network than a tree [147, 155].

Studies not directly related to sequence comparisons include the reconstruction of phylogenetic trees based on the presence or absence of orthologous genes (i.e. overall gene content) [156, 157, 158] and conservation of gene order [148, 158]. Trees based on the former approach creates a distance-based phylogeny by comparing the number of genes a pair of species have in common divided by their total number of genes [157]. More recently, Kunin *et al.* introduced a method called genome conservation which combines sequence similarity and gene content information [159]. In spite of extensive HGT, all these methods still produce phylogenetic trees that are remarkably similar to 16S rRNA trees [156, 157, 158]. This indicates that the systems-level organisation of genes can also be successfully exploited to derive the evolutionary history of organisms. Furthermore, depending on the method employed, the gain in information regarding phylogenetic relationships may be used to enhance the phylogenetic signal from single gene phylogenies [160].

5.1.3 Metabolism-based phylogenies

As discussed in Chapter 4, the availability of complete genomes for hundreds of species has, in turn, allowed the development of online databases which archive organism-specific metabolic repertoires [8, 9]. The metabolome[8] of each organism varies due to the different complement of enzymes encoded in the genome. Phylogenetic trees reconstructed from metabolic data have been referred to as "phylophenetic" since they represent phenotypic features derived from heritable characters [161, 162]. Therefore, comparison of the metabolism between species may be used as a complementary approach to gene-based phylogenies for understanding evolutionary relationships, environmental pressures (e.g. pathogenicity) and to guide metabolic engineering studies. Several groups have applied phyloge-

[7] occurs when an organism receives genetic material from its ancestor.

[8] represents the collection of all metabolites in a biological organism, which are the end products of its gene expression.

netic analyses to metabolic pathways, based on information from:

- *Individual pathways*
  Forst and Schulten [163, 164], were one of the first groups to use metabolic pathway information as a basis for the reconstruction of phylogenetic trees. They introduced a new method with which to calculate the distance between organisms based on a combination of enzyme sequence information and pathway topology. Comparison of network topologies was carried out by Heymans and Singh [165] using an exclusive graph theoretical approach (Section 1.6.1.1). An initial set of 80 organisms was divided according to different criteria and, the distance between organisms was calculated from the network topology of individual and multiple pathways involved in glycolysis, the citric acid cycle, and, carbohydrate and lipid metabolism. The resulting phylogenetic trees were found to be similar to the conventional 16S rRNA-based phylogeny and organisms from the three domains of life were found in separate clusters.

- *Presence or absence of entire pathways*
  Liao *et al.* [166] compared the entire metabolic repertoire of organisms by representing the presence or absence of individual pathways as a binary vector (i.e. a string of zeros and ones). Organisms were then hierarchically clustered according to their metabolic profiles and their relative placement was examined in the corresponding 16S rRNA-based tree. Amongst their findings was the separation of organisms belonging to the archaeal domain according to their kingdom lineage, namely Euryarchaeota and Crenarchaeota.

- *Enzyme content*
  Ma and Zeng [167] reconstructed phylogenetic trees from the enzyme, reaction and gene contents of the entire metabolic networks of 82 organisms. As Liao and colleagues [166] had done with pathways, they represented the enzyme content (i.e. obtained from the KEGG LIGAND database) as a binary string. The distance between pairs of organisms was subsequently calculated using the Jaccard index [168] as a measure of similarity to render phylogenetic trees. As with most other studies introduced thus far, the overall similarity between the obtained phylogenies and 16S rRNA-based trees was remarkably high. A similar study compared the enzyme content from 69 metabolic pathways for 27 organisms representing the three domains of life [162]. Using obligate pathogens as an example, the authors concluded that organisms that were found to be closely related from conventional 16S rRNA-based phylogenies could be distantly related metabolically and vice versa.

- *Reaction content*
  A recent publication by Hong *et al.* [169] reconstructed the metabolic pathway reaction content for 42 microorganisms using databases such as KEGG [8] and MetaCyc [9]. The overall metabolic pathways were divided into 64 subpathways based on the clusters of orthologous groups database[9] (COG†) division [170] of the National Center for Biotechnology Information. Phylogenetic trees were reconstructed by comparing organism-specific profiles of subpathway contents. When compared to 16S rRNA-based trees, the results from the study indicated a good separation of organisms between the three domains of life and the close relationship between eukaryotes and archaea at the level of metabolic networks.

- *Purely graph theoretical*
  Some of the most recent studies that aim to derive phylogenetic trees from metabolic information employ graph theory [171, 172, 173] (Section 1.6.1.1). Liu *et al.* [172] reconstructed enzyme-specific profiles for different organisms and determined their topological importance using three network indices. They concluded that phylogenetic profile is not independent of enzyme network importance and that it correlates better with degree[10] and betweenness centrality[11], but less so with closeness centrality[12]. A similar study employed a strict graph-theoretical approach and applied it to 11 single celled organisms [171]. They used a more extensive set of network measures than the previous study to test whether the intrinsic network design principles are the same amongst the three domains of life. Forst *et al.* [173] represented prokaryotic metabolic networks as directed hypergraphs and used set algebraic measures (i.e. union, intersect and difference) for both the pairwise comparison of networks and identification of distinct metabolic features.

Most of the comprehensive studies highlighted above were carried out with the aim of reconstructing the tree of life from a metabolic perspective and, subsequently, comparing it with the corresponding 16S rRNA-based trees [162, 166, 167, 169]. A downside of this approach is that relatively few species were selected as a representative subset for the different domains within the tree of life. To overcome these limitations, the scope of the work reported here encompasses not only the largest dataset of organisms[13] employed thus far, but also separate classifications for several large taxonomic groups based on enzyme complement (see next

[9] provides phylogenetic classification of proteins encoded in complete genomes.
[10] the number of connections of a network node.
[11] measures how frequently a node appears on all shortest paths between two other nodes.
[12] measures how close a node is to others.
[13] all of which are prokaryotes.

section). A comparison of the resulting phylogenetic trees with 16S rRNA-based trees is then made to determine interesting phenotypic and taxonomic discrepancies. Whatever the data source, the phylogenetic tree reconstruction process involved three primary steps (Section 2.4):

1. Data acquisition via biological databases.
2. Pairwise comparisons between organisms and distance matrix calculation.
3. Data clustering.

The following sections will elaborate on the differing resources and methods employed at each step of the above process whether for enzyme complement or 16S rRNA phylogenies.

5.2 Clustering by enzyme complement

At the time of writing, the KEGG database contained by far the most comprehensively available organism dataset, with metabolic pathway information for 876 organisms (148 Eukaryotes, 676 Bacteria, 52 Archaea) [8] - partially or fully sequenced. Organism-specific files containing gene-EC relationships (files of type '*org*_enzyme.list'[14]) were downloaded and archived locally from the KEGG GENES database (Release 42.0, 1^{st} April 2007). Additional files with extension '.nuc' containing the available complement of ORFs for individual organisms were also downloaded. The initial organism set included 462 prokaryotes with more than 100 fully qualified EC numbers and an available '.nuc' file. An independent list of EC numbers from that of the KEGG annotation was predicted by applying the PRIAM methodology [117] on the ORFs contained in the organism-specific '.nuc' files (Section 4.3.2). Gene-EC data for the organism set was also downloaded at a later date from a more recent version of KEGG GENES (Release 47.0, 1^{st} July 2008). The 462 prokaryotes were then divided into groups according to their taxonomic group, the largest 6 of which were selected for further study:

1. Archaea (34)
2. α-proteobacteria (57)
3. β-proteobacteria (42)
4. γ-proteobacteria (106)
5. Firmicutes (93)

[14] where *org* is a three letter KEGG organism abbreviation (Appendix E).

6. Actinobacteria (37)

Prokaryotes were chosen for three reasons. Firstly, due to a reasonably well-understood metabolism, enzyme functions in individual species can be more reliably identified. Secondly, prokaryotes comprise of two of the three domains of life with the largest number of species that have completed genome sequences, thereby making cross-comparison at the metabolic-level more informative. Finally, limiting the study to particular groups of prokaryotes, as listed above, may be exploited by focusing on metabolic peculiarities within closely related sets of organisms, as opposed to distantly related ones that are likely to be metabolically disparate anyway.

Although, general pathway databases such as KEGG archive organism-specific sequence and metabolic information, the completeness and reliability of this data has been questioned (see Chapter 4 and Appendix F for [134]). One of the limitations of this study is that a relatively small number of prokaryotic species in the KEGG database have fully annotated genomes. Therefore, the enzyme complement of some organisms may not be complete, ultimately, leading to biased comparisons between species. To reduce this bias, instead of comparing the data between different databases and choosing the most comprehensive resource, the EC data obtained from the more recent KEGG GENE release was pooled with the PRIAM output for each organism. However, it is worth noting that even in the combined dataset, there will still be some, or even many, undefined enzymatic activities. The advantage gained by pooling EC data for improving the comparative accuracy between organisms will be discussed in Section 5.5.

By consulting databases that encode the relationships between different biological entities (e.g. KEGG LIGAND), the reaction complement of an organism can be deciphered from its EC complement (Section 4.3). Thereafter, metabolic trees could have also been reconstructed from organism-specific reaction profiles [169] instead of, or complementary to EC-level phylogenies. The downside of this method is that a number of reactions may be associated with a given EC number, not all of which are necessarily catalysed by every organism's gene product [129]. Without further species-specific information regarding enzyme-reaction relationships it would be expected that little advantage would be gained by comparison of the EC or reaction sets between organisms. In accordance with similar work carried out by Ma and Zeng [167], the comparison of EC- and reaction-based metabolic trees for all the prokaryotic groupings revealed very few, if any, differences (results not included). To reduce the level of uncertainty associated with enzyme-reaction relationships for different organisms, all metabolic trees were reconstructed using EC complement information.

5.2.1 Data clustering

The primary aim of this study is to hierarchically cluster (Chapter 2) organisms at the metabolic-level by comparing their complement of enzymes. The underlying assumption is that different species have different degrees of metabolic commonality between them. For a given organism, the EC complement was obtained from the union of EC numbers obtained from the PRIAM output and the more recent version of KEGG[15]. The next step was to calculate a symmetric distance matrix ($\mathbf{D}_{ii}$) using the `RowDiffMtx` function (Section 2.4.3). Before $\mathbf{D}_{ii}$ can be calculated, an appropriate measure must be selected to calculate the pairwise distance between the enzyme profiles for pairs of organisms. By analogy to the definition reported by Korbel *et al.* [158] for comparison of gene content between species, a weighted distance measure will be applied to compare EC complement. Ma and Zeng [167], referred to this distance measure as the 'Korbel' distance and compared it to standard measures such as the Jaccard index[16] which is more susceptible to larger differences in EC complement between organisms. The Korbel distance between a pair of organisms is defined as the number of ECs in common divided by the weighted average union of their EC complement:

$$d_{ij} = 1 - \frac{n_{ij}}{\sqrt{2}n_i n_j / \sqrt{n_i^2 + n_j^2}}$$

where n_i and n_j are the number of ECs in each of the organisms to be compared and n_{ij} is the size of their intersect. Therefore, two organisms with identical EC datasets will have a Korbel similarity measure of 0 and those with nothing in common will have a value of 1. From the computed distance matrix, all phylogenetic trees were generated in Newick format using an hierarchical agglomerative clustering algorithm implemented in ScrumPy. The MEGA software package was used to visualise and edit all the phylogenetic trees.

5.3 Obtaining 16S rRNA phylogenies

16S rRNA sequences for all organisms in the set were also obtained in order to compare the topology of the metabolic trees with those of the accepted standard. 16S rRNA sequence information for each of the 462 organisms was automatically downloaded from the 'Integrated Microbial Genomics' section of the DOE Joint Genome Institute†, along with supplementary taxonomic and phenotypic data. Taxonomic

[15] unless stated otherwise all EC complement trees will refer to those that used the latter KEGG release as opposed to the older one.

[16] the size of the intersection divided by the size of the union of the sample sets.

information for individual species included the following ranks (from highest to lowest): domain, phylum, class, order, family, genus, and species. Organism-specific phenotypic descriptions such as ecotype, oxygen requirements, gram stain and disease were also obtained from the NCBI Taxonomy website†.

The presence of multiple, heterogenous rRNA operons, especially in bacterial species, complicated the sequence selection procedure for the reconstruction of 16S rRNA-based trees [174]. For example, 30 16S rRNA sequences were available for *Clostridium perfringens* SM101, 20 with a length of 1616bp, 9 with a length of 1522bp and 1 with a length of 1517bp. To overcome this shortcoming, if an organism had more than one 16S rRNA sequence, then the most frequently occurring one was chosen, otherwise the most frequent and/or shortest one was selected. Therefore, from this criteria a sequence is more likely to be representative of a species based on its frequency of occurrence, otherwise the shortest sequence(s) must, in theory, carry the same amount of information as longer sequences. Using customised tools written in Python, a single 16S rRNA sequence was automatically selected for all the prokaryotes within a taxonomic group and written to file in FASTA format† [175].

5.3.1 Data clustering

There are a number of programs available for rendering phylogenetic trees from the multiple sequence alignment of molecular data [137]. The most popular of these are CLUSTALW† [176], T-COFFEE† [177] and, the more recent MUSCLE† [178] programs. Although, T-COFFEE is amongst the most accurate multiple sequence alignment methods, at any one time its web server can only process 50 sequences with a maximum length of 2000bp. The γ-proteobacteria group contained 106 species, therefore, the use of T-COFFEE was not a practical option. The MUSCLE tool was chosen for the purposes of this study since it has been reported to be faster and more accurate than CLUSTALW when compared to reference alignments [178]. The files in FASTA format for each taxonomic group were pasted into the MUSCLE web server interface on the EBI website†, and executed with default settings. MUSCLE uses a progressive alignment technique which initially aligns the most similar sequences first and then progressively adds the more dissimilar sequences to build the alignment. A distance matrix is subsequently rendered whereby distance can be roughly defined as the percent sequence difference between all the possible pairs of sequences. The distance matrix is used to reconstruct the phylogenetic tree using the UPGMA method (Section 2.4.2) and outputted from MUSCLE in Newick format.

Taxonomic group	Similarity Measure	
	KEGG Release 42.0	KEGG Release 47.0
α-proteobacteria	0.261	0.216
β-proteobacteria	0.309	0.222
Archaea	0.262	0.246
Actinobacteria	0.282	0.268
γ-proteobacteria	0.297	0.273
Firmicutes	0.339	0.311

Table 5.1 – TREEDIST-derived symmetric differences between the phylogenetic trees reconstructed from EC complement and 16S rRNA sequences, for the six taxonomic groups used in the study. EC complement was calculated using two releases of KEGG and integrated with the same PRIAM output. Identical topology and maximum possible difference between trees would be indicated by values of 0 and 1, respectively.

5.4 Comparison of EC complement and rRNA phylogenies

For each of the six taxonomic groups, the phylogenetic trees derived from EC complement and 16S rRNA will be presented alongside each other. Unique KEGG organism abbreviations (Appendix E) were used on the leaves of the trees for ease of comparison, and to avoid the confusion associated with the same species being sequenced by different groups. Only the overall tree topology will be considered since the distance scales between the two types of trees (i.e. based on comparison between gene sequences and overall EC complement) are not directly comparable. The TREEDIST[†] program of the PHYLIP package (version 3.66), which uses the Symmetric Distance algorithm described by Robinson and Foulds [179] was used to derive quantitative measurements for the topological similarity between each pair of trees. If a partition in the tree is a branch dividing the set of organisms in two sets into two groups (those connected to one end of the branch and those connected to the other), the Symmetric Distance is the number of partitions that are present in one tree but not the other. Using the method described by Aguilar *et al.* [162], the Symmetric Distance was divided by the maximum possible number of internal branches ($4n$-6 for n species). This allowed the results to be scaled between 0 (identical topology) and 1 (maximum topological difference).

5.5 Results and discussion

From the results in Table 5.1, it is evident that the α-proteobacteria group was found to be topologically closest to its corresponding 16S rRNA-based tree; symmetric difference of 0.216. On the other hand, the largest symmetric difference

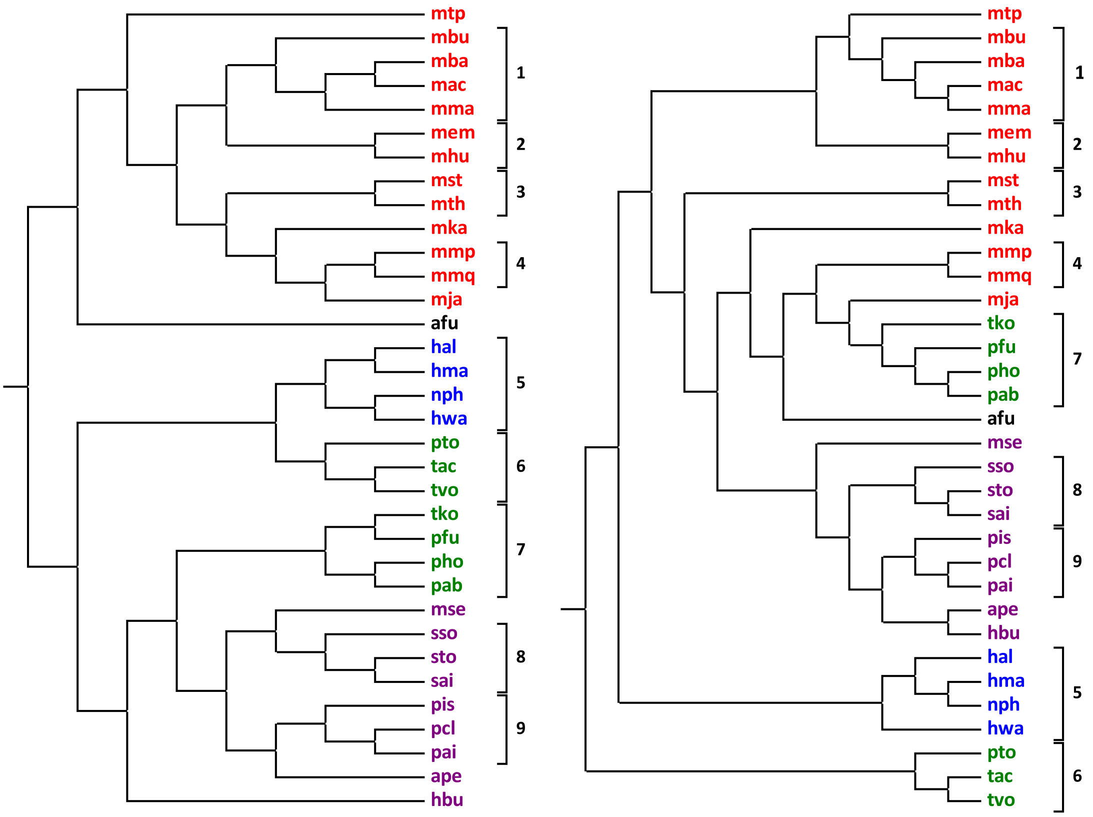


Figure 5.1 – Side-by-side cladogram illustration generated according to the EC complement (left) and 16S rRNA sequence information (right) for 34 organisms in the archaeal set. Amongst the Euryarchaeota phylum, methanogens are shown in red, halophiles in blue and thermophiles in green. The Crenarchaeotan phylum, all of which are thermophilic species are highlighted in purple. Note that the Euryarchaeotan species, *afu* is the only species from the *Archaeoglobaceae* family included in this study, and, although it has thermophilic properties it has not been highlighted in any of the groups described above. Numbered brackets indicate clusters that contain the same species in both trees. See Appendix E for full names of organisms.

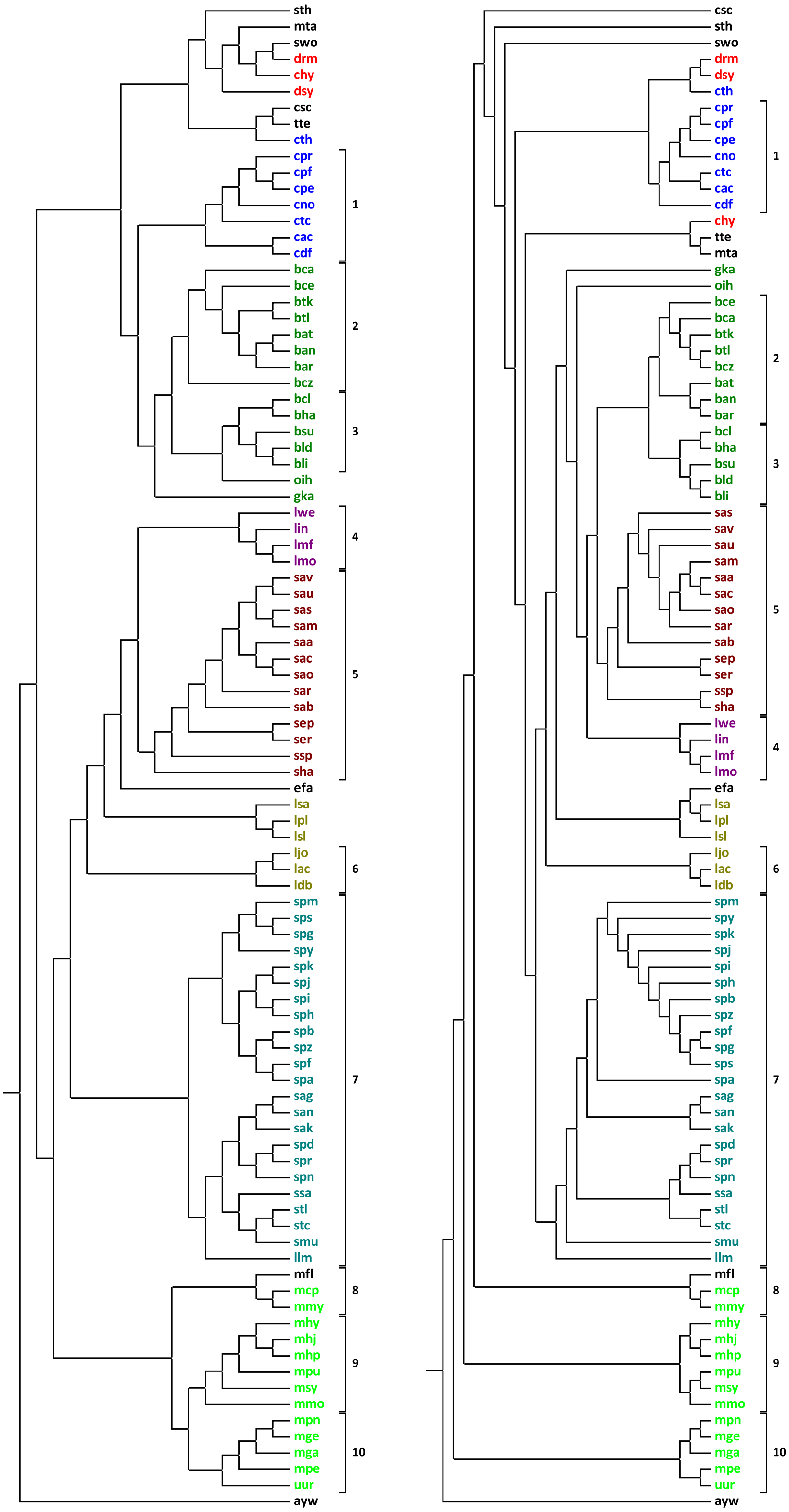


Figure 5.2 – Side-by-side cladogram illustration generated according to the EC complement (left) and 16S rRNA sequence information (right) for 93 organisms in the firmicutes set. For ease of interpretation the families containing three or more species have been coloured; *Peptococcaceae* in red, *Clostridiaceae* in blue, *Bacillaceae* in green, *Listeriaceae* in purple, *Staphylococcaceae* in maroon, *Lactobacillaceae* in olive, *Streptococcaceae* in teal and *Mycoplasmataceae* in lime. Numbered brackets indicate clusters that contain the same species in both trees. See Appendix E for full names of organisms.

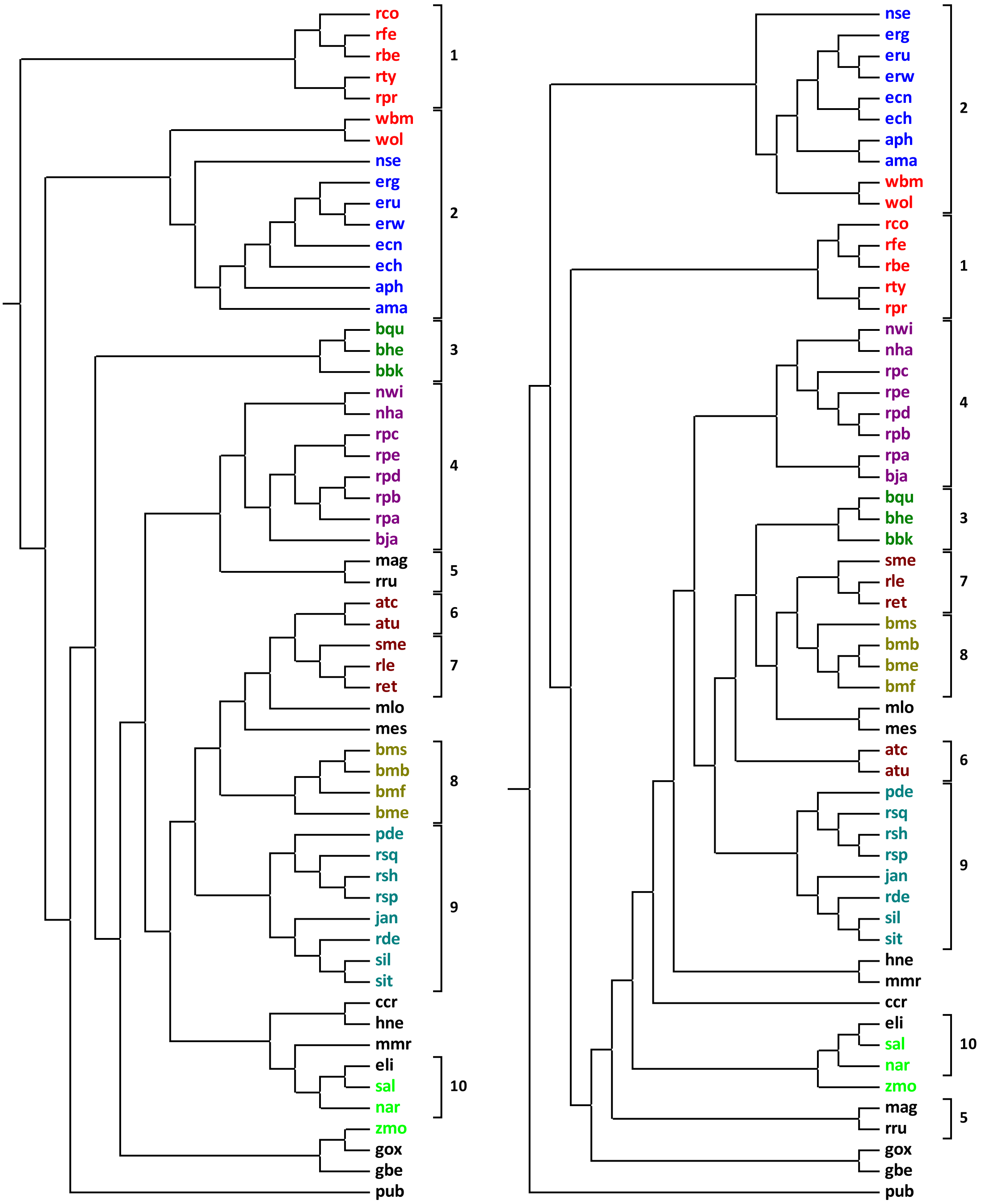


Figure 5.3 – Side-by-side cladogram illustration generated according to the EC complement (left) and 16S rRNA sequence information (right) for 57 organisms in the α-proteobacteria set. For ease of interpretation the families containing three or more species have been coloured; *Rickettsiaceae* in red, *Anaplasmataceae* in blue, *Bartonellaceae* in green, *Bradyrhizobiaceae* in purple, *Rhizobiaceae* in maroon, *Brucellaceae* in olive, *Rhodobacteraceae* in teal and *Sphingomonadaceae* in lime. Numbered brackets indicate clusters that contain the same species in both trees. See Appendix E for full names of organisms.

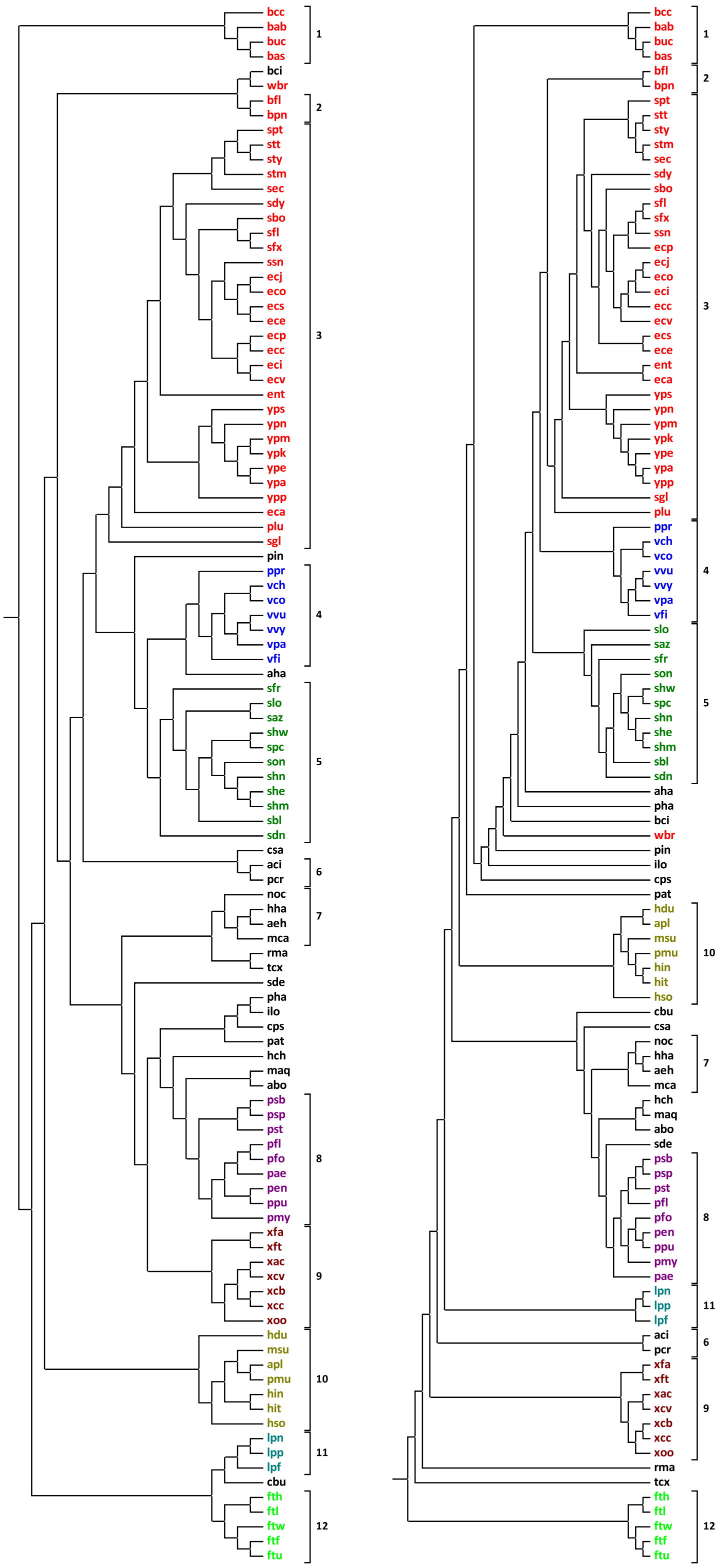


Figure 5.4 – Side-by-side cladogram illustration generated according to the EC complement (left) and 16S rRNA sequence information (right) for 106 organisms in the γ-proteobacteria set. For ease of interpretation the families containing three or more species have been coloured; *Enterobacteriaceae* in red, *Vibrionaceae* in blue, *Shewanellaceae* in green, *Pseudomonadaceae* in purple, *Xanthomonadaceae* in maroon, *Pasteurellaceae* in olive, *Legionellaceae* in teal and *Francisellaceae* in lime. Numbered brackets indicate clusters that contain the same species in both trees. See Appendix E for full names of organisms.

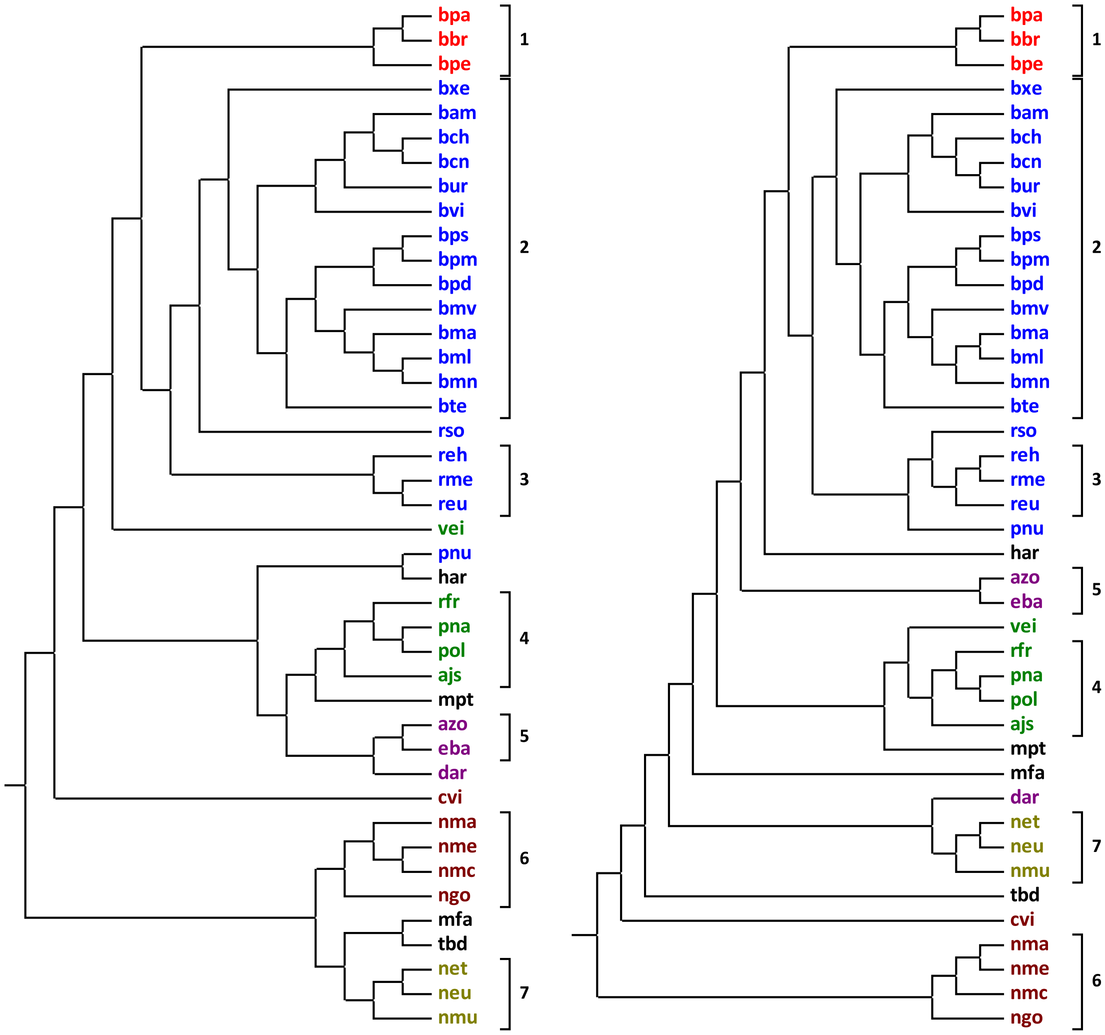


Figure 5.5 – Side-by-side cladogram illustration generated according to the EC complement (left) and 16S rRNA sequence information (right) for 42 organisms in the β-proteobacteria set. For ease of interpretation the families containing three or more species have been coloured; *Alcaligenaceae* in red, *Burkholderiaceae* in blue, *Comamonadaceae* in green, *Rhodocyclaceae* in purple, *Neisseriaceae* in maroon and *Nitrosomonadaceae* in olive. Numbered brackets indicate clusters that contain the same species in both trees. See Appendix E for full names of organisms.

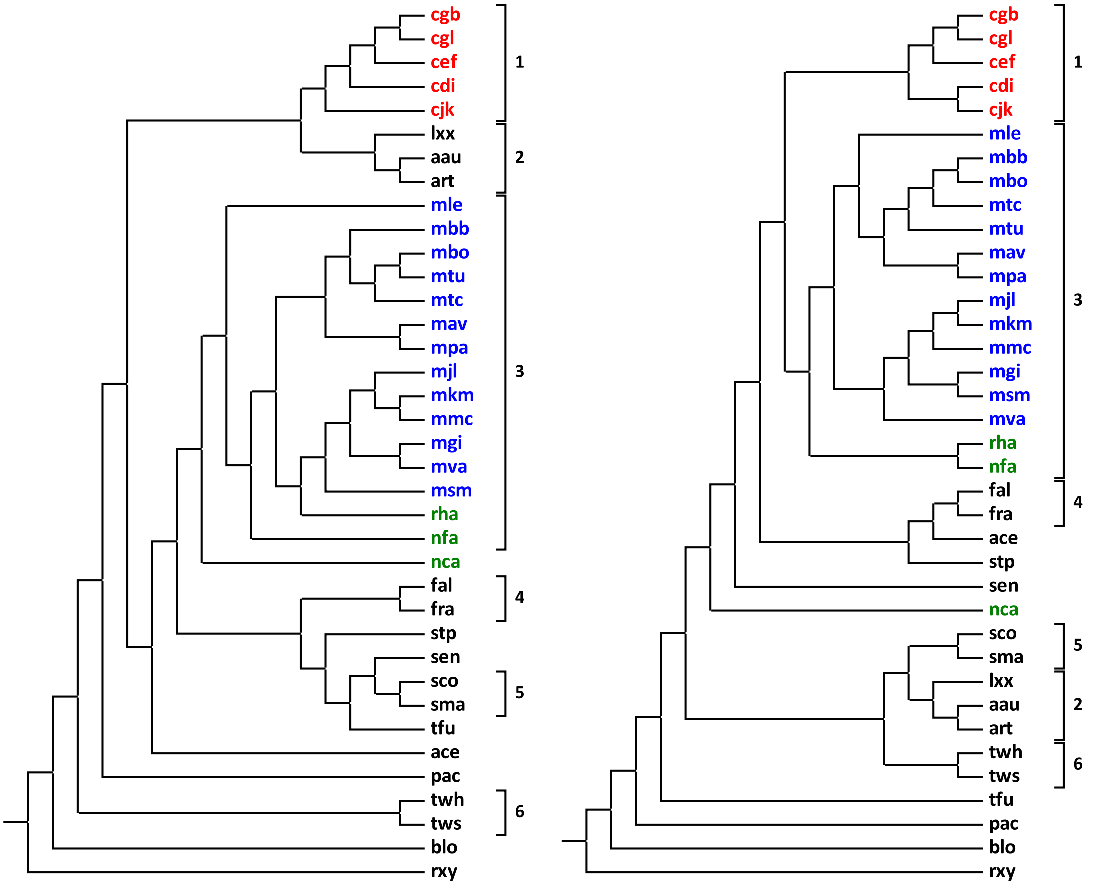


Figure 5.6 – Side-by-side cladogram illustration generated according to the EC complement (left) and 16S rRNA sequence information (right) for 37 organisms in the actinomycete set. For ease of interpretation the families containing three or more species have been coloured; *Corynebacteriaceae* in red, *Mycobacteriaceae* in blue and *Nocardioidaceae* in green. Numbered brackets indicate clusters that contain the same species in both trees. See Appendix E for full names of organisms.

(0.311) was observed for the firmicutes which had the second largest organism dataset (93). As discussed in the next section, all the EC complement trees were found to be less similar to their 16S rRNA counterparts when reconstructed from the older version of the KEGG database. Alternatively, the similarity between trees can be visualised by the relative proximity of clusters containing the same organisms, with further emphasis on those organisms with branching discrepancies (Figures 5.1–5.6). It is worth noting here that, although, clusters containing the same organisms have been highlighted between the corresponding trees in the results, additional organisms may have been included in some groups based on the proximity of their branching patterns. The distribution of the organisms between pairs of trees was analysed with regard to additional species-specific criteria such as phylogenetic proximity, ecotype, oxygen requirements and pathogenicity.

Figure 5.1 depicts the pair of phylogenetic trees computed for 34 organisms in the archaeal domain. The organisms in this taxa are obligate or facultative anaerobes that grow in unusual surroundings such as hydrothermal vents and highly saline environment (e.g. Dead sea). Despite their different habitats, the metabolic tree indicates that archaeal strains have a significant overlap in their enzyme complement, and the degree of this overlap provides a taxonomic classification which is very similar to the 16S-rRNA tree. A good separation between euryarchaeota and crenarchaeotan species was obtained. However, rather than being clustered together in Figure 5.1, one class of thermophilic euryarchaeotan species (Thermococci; Cluster 7) was found to be topologically closer to the crenarchaeotan (Clusters 8 and 9) and the other (Thermoplasmata; Cluster 6) to the Halobacteria (Cluster 5). In contrast to the findings by Ma and Zeng [167], these observations are also visible in the 16S rRNA-based tree whereby Clusters 5 and 6 are found together and Cluster 8 is closest to another group containing Cluster 7. A phylogenetic analysis derived from gene content indicated that the crenarchaeota should instead be clustered with species from Thermoplasmata as opposed to Thermococci, which were preferentially grouped with the methanogens [180] (Figure 5.1; red). The archaeal organism set in this study was roughly twice as large as that used by Ma and Zeng [167], and, therefore permits a more concrete validation since more organisms are shown to follow the same pattern. On further inspection of phenotypic information, the Thermococci in Cluster 7 are able to carry out sulphur respiration in extremely thermophilic environments which is also characteristic of crenarchaeotan organisms. Furthermore, the specialised metabolism exhibited by the halophiles (blue) may be used to account for their proximity to the acidophilic Thermoplasmata. *Archaeoglobus fulgidus* DSM 4304 (*afu*) was an anomaly as it is a potential pathogen (i.e. harms host by production of hydrogen sulphide gas) with thermophilic, piezophilic[17] and sulphur reducing properties. In the metabolic tree it was clustered as a potential outgroup of the methanogens whereas in the rRNA tree it was found closer to the Methanococci (including thermophilic *mja*; Cluster 4) and the thermophilic, sulphur respiring Thermococci. Inclusion of further Archaeoglobi may help resolve the position of *afu* in both trees, and any subsequent phenotypic interpretations.

Prokaryotes with reduced EC complement (ranging from 133–311) were found amongst the firmicutes (*Mycoplasmatales*; Clusters 8, 9 and 10 in Figure 5.2), α-proteobacteria (*Rickettsiales*; Clusters 1 and 2 in Figure 5.3) and γ-proteobacteria (*Enterobacteriales*; Clusters 1 and 2 in Figure 5.4). All of the species within this category were found to be either endosymbionts or obligate parasitic organisms. Their minimal metabolic repertoires can be explained by the loss of genes that become unnecessary for survival in a nutrient-rich environment provided by the host organism [169]. The placement of the parasites within the *Mycoplasmatales* family were in almost perfect agreement with the 16S rRNA-based tree reconstructed as part of this study (Figure 5.2) and those created independently [181]. Similar results were also obtained for the *Rickettsiales*, whereby two distinct but closely associated clusters were found for the *Rickettsiaceae* and *Anaplasmataceae* families (Figure 5.3). In both the metabolic and rRNA-based trees, the $\gamma-proteobacteria$ in the *Buchnera* genus were found in topologically identical clusters of their own (Cluster 1; Figure 5.4). However, this was not the case for the remainder of the endosymbionts within this taxonomic group. The EC complement phylogeny showed a consistent grouping of the two organisms within the *Candidatus blochmannia* genus (Cluster 2; Figure 5.4) and two other endosymbionts (*wbr* and *bci*). Within the corresponding 16S rRNA-based tree, the latter two organisms were found in clusters of their own albeit topologically adjacent to each other. The NCBI Taxonomy page for *Baumannia cicadellinicola* Hc (*bci*; Taxonomy ID: 374463) indicates that the lineage of the organism is unclassified beyond the class-level. Although further evolutionary-based research may be required to determine its full lineage, the results from the metabolic tree may be used to suggest that it is either part of, or closely related to the *Enterobacteriales* order and *Enterobacteriaceae* family. Furthermore, once fully resolved the phylogenetic position of *bci* may be found to be in the immediate proximity of the organisms indicated by the EC complement tree.

Other metabolism-based phylogenies have reported the grouping of parasites and symbionts from various taxa within the three domains of life [162, 167, 169]. For example, Aguilar *et al.* found that obligate parasites are clustered together de-

[17] also called a barophile, is an organism which thrives at high pressures, such as deep sea bacteria or archaea.

spite their differing placements in the universal tree of life, possibly as a result of a convergence of their metabolism due to a shared lifestyle. In conjunction with the results from this study, Ma and Zeng [167] found that a pair of γ-proteobacteria, *Buchnera aphidicola* Sg (*bas*) and *Buchnera aphidicola* APS (*buc*) were preferentially clustered according to their genetic relatedness as opposed to with other symbionts within the three domains of life. The larger organism set used in this study and the investigation of individual taxa permits a stronger assessment of this claim. Within taxonomic groups, parasites and endosymbionts show a remarkably similar clustering profile when comparing phylogenies reconstructed from independent sources of biological information. This outcome provides a direct validation for the use of the Korbel distance measure by proving its effectiveness for separating organisms with smaller EC complement. In summary, the metabolic specialisation observed for parasitic and endosymbiont species becomes more evident whilst surveying organisms from different eubacterial taxa, but, at the class-level this specialisation is more reflective of their phylogenetic origins.

With the exception of the archaea (potential pathogen *afu*; *Archaeoglobus fulgidus* DSM 4304), pathogenic organisms were scattered throughout the chosen taxonomic groups. Including parasites and selected endosymbionts, the downloaded organism-specific phenotypic information indicated that 65.5% of the firmicutes, 53.8% of the γ-proteobacteria, 45.2% of the β-proteobacteria, 43.2% of the actinomycetes and 38.6% of the α-proteobacteria were pathogenic species. Overall the clustering results indicated that the pathogenic phenotype had little or no influence on the grouping of eubacterial species. For example, the β-proteobacteria in the *Burkholderia* genus (Cluster 2 in Figure 5.5) has 14 pathogenic and 2 non-pathogenic species. The topological distributions of these bacteria were identical in both the metabolic and rRNA based trees. Moreover, the two non-pathogens *Burkholderia pseudomallei* 1710b (bpm) and *Burkholderia thailandensis* E264 (bte) were found in separate clusters amongst the pathogenic organisms. Other examples of the non-uniform distribution of pathogens was observed amongst the *Bacilli* (Clusters 2 and 3) and *Streptococci* (Cluster 7) for the firmicutes (Figure 5.2). Based on these findings, a possible explanation is that any metabolic specialisation in pathogenic species has minimal influence on their inherited complement of metabolic enzymes.

In general, species belonging to the same taxonomic family are, in most cases clustered together in the EC complement phylogenies. However, a few exceptions were found which have similar ecotypic and metabolic activities to species from other families. *Zymomonas mobilis mobilis* ZM4 (*zmo*) clustered correctly with the remainder of the *Sphingomonadales* within the α-proteobacterium rRNA tree (Cluster 10 in Figure 5.3). On the other hand, based on EC complement it was found to be closer to a pair of species from the *Rhodospirillales* order, namely, *Gluconobacter oxydans* 621H (*gox*) and *Granulobacter bethesdensis* CGDNIH1 (*gbe*). On further inspection of the ecotypic properties of the organisms in this cluster, *zmo* and *gox* are both industrially important fermentative prokaryotes found in plants, whereas *gbe* is a novel pathogen associated with human chronic granulomatous disease. In the β-proteobacteria rRNA tree, *Verminephrobacter eiseniae* EF01-2 (*vei*) was also found amongst members of its own family (*Comamonadaceae*; Cluster 4 in Figure 5.5) which was not the case in the EC complement tree. It was in an single organism cluster in between its own family members and three clusters that contained mostly pathogenic species (Clusters 1–3). In contrast to the other endosymbionts discussed so far, *vei* had a particularly large repertoire of enzymes (830 ECs), possibly stemming from its colonisation of juvenile earthworms (*Eisenia foetida*) during embryonic development. The other members of the *Comamonadaceae* are metabolically specialised organisms with potential bioremediation applications for the degradation of various pollutants such as naphthalene (*Polaromonas naphthalenivorans* CJ2; *pna*), cis-dichloroethane (*Polaromonas sp.* JS666; *pol*) and 2-nitrotolulene (*Acidovorax sp.* JS42; *ajs*). Interestingly, *Methylibium petroleiphilum* PM1 (*mpt*) has an unclassified lineage at the family-level, and its close association to the *Comamonadaceae* in both EC complement and rRNA trees may be a potential indicator as to its full taxonomic identity.

Outlier organisms can be thought of as those that are in a cluster of their own and behave as a closely-related version for the root of a tree. The two actinomycetes (Figure 5.6), *Bifidobacterium longum* NCC2705 (*blo*) and *Rubrobacter xylanophilus* DSM 9941 (*rxy*), as well as the firmicute *Aster yellows witches-broom phytoplasma* AYWB (*ayw*) were noted as outlier species in both rRNA and EC complement trees. Although the latter organism was found to have the smallest number of ECs (133) in the entire dataset, a possible explanation for these observations is that all of these species were the only representatives from their order-level lineage. The NCBI Genome Project description for *rxy* states that it represents the oldest lineage of the Actinobacteria and that it is distantly related to the *Mycobacteria* and *Streptomyces*, which is also in accordance with the results in this study (Clusters 3 and 5 in Figure 5.6, respectively). The fact that *rxy* is an outlier in the EC complement tree demonstrates that deeper phylogenetic branches can also be obtained by metabolic comparisons. Alternatively, the inclusion of additional species may be used to further resolve the positions of outlier species in relation to the remainder of organisms in the tree.

As discussed in Chapter 4, owing to a select few organisms which have been extensively annotated (e.g. *E. coli*), most organisms are automatically assigned enzymatic activities based on functional homology. In some cases, sequence similarity may not imply the same function and certain enzymatic activities may go unnoticed due to the evolutionary divergence between sequence information. For the purpose of this study, the justification for the use of EC complement stems from the fact that the comparison of other metabolic entities such as biochemical reactions will introduce further uncertainty, primarily from the differing enzyme specificity between species. From Table 5.1, it is evident that the EC complement trees reconstructed using the newer release of KEGG is topologically closer to the 16S rRNA-based phylogeny. Furthermore, a larger symmetric difference would have been obtained if the data from either KEGG or PRIAM was used. Despite a significant advantage being gained by pooling the EC information using KEGG and PRIAM, it may be assumed that there are still a number of enzymatic activities that remain undiscovered, more so for some organisms than others. Consequently, a level of uncertainty is introduced within the results which is, so far, unavoidable and will only improve with increased curation and annotation efforts. Regardless of the quality of data used, it is evident that the phylogenetic trees reconstructed from EC complement still reflect the accepted standard phylogeny, with additional metabolic discrepancies.

CHAPTER 6

General Discussion and Future Directions

The relatively simple data requirements for structural modelling techniques may account for their widespread use. Using this approach the structure of the system is considered, which is the most basic feature of any network. As discussed later, another contributing factor is the ability to build structural models to investigate the properties within organism-specific metabolic networks on a genome-scale. To obtain an insight into the functioning of such networks a number of structural analytical techniques are available. Unfortunately, elementary modes analysis (EMA) cannot be applied to genome-scale models due to problems with computational complexity. Nevertheless, the potential applications of EMA demonstrate that it is still worth using this method to investigate smaller models or sub-networks where feasible. However, the subsequent interpretation of the datasets obtained from EMA poses many challenges. As part of this project, hierarchical clustering techniques were used to indicate the usefulness of further analytical procedures for the interpretation and illustration of the datasets obtained from EMA. The purpose of Chapters 2 and 3 was not to survey the multitude of techniques that could potentially be used to cluster EMs datasets, but rather to highlight how such methods may be useful for their analysis, especially for much larger datasets in comparison to those employed in this study. The results indicated that there are no strong reasons to prefer clustering of EMs by reaction profile over clustering by net stoichiometry, and a better outcome is obtained using the two approaches in combination. Within the general cluster patterns lie potentially important characteristics regarding similarities in profiles for modes in a single cluster. Furthermore, the parallel use of coloured matrix visualisation methods can be used to provide a quantitative and qualitative reflection of the original theoretical observations. This sort of exploratory analysis may be able to provide a simple and intuitive description of clusters which is easy for human comprehension, ultimately, leading to an insight into modes with common metabolic functions.

It has taken less than 12 years, from the publication of the 1^{st} bacterial genome in 1995 (*H. influenzae* Rd [182]) to the predicted completion of the 1000^{th} microbial genome by 2008 [183]. In accordance with this surge in biochemical information, the stoichiometric data required for the reconstruction of genome scale, organism-specific structural models is becoming more readily available. Databases such as KEGG and BioCyc provide a comprehensive catalogue for biochemical data, and also offer the possibility of automatically reconstructing whole metabolic models of specific organisms directly. Consequently, the acquisition of reaction data from pathway databases, and its subsequent translation into a format suitable for modelling has become technically trivial. The major difficulty herein lies in the quality of the resulting metabolic network and its compromised ability to reflect the real organism properties at the systems-level. As indicated in Chapter 4, for small hand-built models, errors of the type reported would rarely be present. In contrast, for large automatically generated models the treatment of these problems becomes more challenging and requires extensive manual curation. Metabolic networks created in such a manner are far from being complete and require verification and refinement before they are suitable for modelling purposes. Once satisfied with the model, the modeller can only assume that the most relevant properties have been accurately captured. To deal with uncertainty, from the perspective of the modeller there are 'identifiable' and 'unidentifiable' components to reconstructing large automated networks. At any given time, the former includes all that is possible to know about the metabolism of an organism of interest, the entirety of which is very unlikely to be included in an initial database-derived model, and the latter, as indicated by the name, is a consequence of insufficient or undiscovered knowledge. For example, our study indicated that approximately half of the reactions in all genome-scale models are associated with orphan metabolites. For the most part, the affect that orphan metabolites have on this type of analysis is a symptom of, as yet, unknown reactions and their contribution to the network properties can only be ignored or made peripheral. Other than the organism-specific enzyme and reaction complements that can be readily obtained from databases, the majority of automated reconstructions contain little information regarding transporters (since they are not associated with an EC number) and their associated compartment information, reversibility criteria and enzyme specificity. At present, there is no standard database available that links all of this biochemical information in a standardised and curated manner, and in most cases these issues can only be addressed by parsing a variety of databases and/or manual literature searching. With reference to the latter solution, an interdisciplinary field called text mining is being used to obtain specific information directly from the literature. However, this is limited by the availability of electronic journals, the possibility of obtaining data in numerous different formats and the vast number of publications that potentially have to be searched [184]. To detect and deal with model reconstruction problems, a combination of local data processing, and the implementation of more robust analysis algorithms and web-based error reporting interfaces seem to be the most promising solutions. In the long-term, it may be necessary to implement an efficient synchronisation between genome annotation and *in silico* modelling efforts with high-throughput experimental research, and the resources that archive them. Due to network size and complexity, one of the problems is to distinguish between genuine differences in a metabolic network, and apparent differences because of incomplete or incorrect annotations of the genome sequence. Taken together, the methods introduced in this study, will be applied to assist the genome annotation process and to investigate the metabolic network variations that biochemically define a given organism. At present, it seems that the effort required to 'clean-up' large metabolic models far outweighs the potential gain in biological knowledge. The previously proposed level of integration between experimental, theoretical and computational techniques can only be of use if they are all designed with these applications in mind.

In Chapter 5, the phylogenetic studies carried out based on enzyme complement also suffer from some of the automated reconstruction issues discussed above. Organism-specific data was derived directly from enzyme information without the need for the reconstruction of structural models. Therefore, the main source of error would have propagated into the study from the database-level since those at the systems-level were not considered. Amongst all the possible metabolic enzymes that could exist *in vivo* a considerable amount still remain to be discovered or are associated with a high degree of uncertainty (e.g. incomplete EC numbers such as 1.1.-.-). With regard to the latter point the PRIAM methodology may be suitable for finding fully qualified versions of the incomplete ECs reported by KEGG. This demonstrates the usefulness of using more specific enzyme profiles for the functional annotation of genes. The EC complement trees were limited in their ability to generate phylogenetic predictions by the amount and quality of data included for individual species. The availability of more reliable information will help elucidate the positions of both the anomalous and well-defined organisms in the enzyme-based trees. Nevertheless, the agreement of the enzyme complement trees with their rRNA-based counterparts implies that rough representations can still produce informative results, with some interpretive and predictive potential. Further research can be carried out by building phylogenetic trees using reaction or pathway information but with the prospect of introducing further problems. For example, not all the reactions attributed to a particular EC number may be catalysed by every organism's corresponding gene product. It follows that the total number of reactions per organism will in fact be an overestimate of its actual metabolic capabilities.

Irrespective of the data source (i.e. genomic or metabolic data) the deeper branching patterns within phylogenetic trees still remain mostly unresolved. More specifically, using EC and rRNA information there was a remarkable similarity in the clustering patterns between organisms at the tips of the trees. In most cases, the same cannot be said about the branching patterns that unite the various taxonomic families and, the preceding higher-order ranks. This may imply that the phylogenetic signal within even closely related organisms has faded over the course of evolution. Some researchers (more notably Doolittle and colleagues [147]) have proposed that extensive horizontal gene transfer may be responsible, with additional arguments stating that the tree of life may be better represented as a network, at least for prokaryotic species. To increase the phylogenetic signal for deeper evolutionary relationships it will be necessary to integrate the biochemical data from 'omic' studies, or on a rationally selected, substantial part of this information. The future potential of phylogenetic studies will be realised by the progression from a single-gene based to a species-specific cellular perspective.

A rather optimistic future for metabolic modelling lies in the ability to create so-called 'virtual' or 'digital' cells. The time taken for this state of knowledge to be realised depends on numerous factors, the most important of which are the size and complexity of the system under investigation. For example, the small human red blood cell (RBC) network is already well established [14, 15, 16], and has been used to simulate common RBC pathologies [17]. Owing to the complexity of other eukaryotic cells, it seems likely that most of the succeeding whole cell metabolic studies will be of prokaryotic origin. The growth in computer technology will help to resolve the analytical issues with modelling techniques. Unfortunately, let alone the bleak prospect of kinetic modelling, even structural metabolic reconstructions for prokaryotic species are far from being complete. Structural modelling on a genome-scale offers the prospect to investigate the metabolic properties for a given organism, but its scope of discovery is limited by the amount and quality of data that is incorporated. Although, there are a number of methods with which to improve the data included in metabolic reconstructions, nothing can be done to reduce the uncertainty caused by the biological information that is yet to be discovered. To this end, an iterative approach to structural modelling can be applied to help find gaps in the metabolic network, and subsequently, to carry out experimental validation and model updating. Despite this potential, approximately half of the metabolites in genome-scale metabolic networks are only present once, even for well-curated prokaryote databases [134]. The post-genomic era has become renowned for the accumulation of biological data, but ultimately, in the years to come, the significance of this information has to be realised at the systems-level.

References

[1] R. Heinrich and S. Schuster. *The regulation of cellular systems.* Chapman & Hall, London, England, 1996.

[2] J.H. Hofmeyr. Steady-state modelling of metabolic pathways: a guide for the prospective simulator. *Comput Appl Biosci*, 2(1):5–11, Apr 1986.

[3] A.L. Barabasi and Z.N. Oltvai. Network biology: understanding the cell's functional organization. *Nat Rev Genet*, 5(2):101–113, Feb 2004.

[4] S. Schuster, T. Pfeiffer, F. Moldenhauer, I. Koch, and T. Dandekar. Exploring the pathway structure of metabolism: decomposition into subnetworks and application to *Mycoplasma pneumoniae*. *Bioinformatics*, 18(2):351–361, 2002.

[5] T. Dandekar, F. Moldenhauer, S. Bulik, H. Bertram, and S. Schuster. A method for classifying metabolites in topological pathway analyses based on minimization of pathway number. *Biosystems*, 70(3):255–270, Aug 2003.

[6] G.D. Bader, A. Heilbut, B. Andrews, M. Tyers, T. Hughes, and C. Boone. Functional genomics and proteomics: charting a multidimensional map of the yeast cell. *Trends Cell Biol*, 13(7):344–356, Jul 2003.

[7] Y.M. Galperin. The Molecular Biology Database Collection: 2007 update. *Nucleic Acids Res*, 35(Database issue):D3–D4, Jan 2007.

[8] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Res*, 36(Database issue):D480–D484, Jan 2008.

[9] R. Caspi, H. Foerster, C.A. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S.Y. Rhee, A.G. Shearer, C. Tissier, T.C. Walk, P. Zhang, and P.D. Karp. The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Res*, 36:D623–D631, Jan 2008.

[10] I. Vastrik, P. D'Eustachio, E. Schmidt, G. Joshi-Tope, G. Gopinath, D. Croft, B. de Bono, M. Gillespie, B. Jassal, S. Lewis, L. Matthews, G. Wu, E. Birney, and L. Stein. Reactome: a knowledge base of biologic pathways and processes. *Genome Biol*, 8(3):R39, 2007.

[11] P.D. Karp, I.M. Keseler, A. Shearer, M. Latendresse, M. Krummenacker, S.M. Paley, I. Paulsen, J. Collado-Vides, S. Gama-Castro, M. Peralta-Gil, A. Santos-Zavaleta, M.I. Penaloza-Spinola, C. Bonavides-Martinez, and J. Ingraham. Multidimensional annotation of the *Escherichia coli* K-12 genome. *Nucleic Acids Res*, Oct 2007.

[12] K.R. Christie, S. Weng, R. Balakrishnan, M.C. Costanzo, K. Dolinski, S.S. Dwight, S.R. Engel, B. Feierbach, D.G. Fisk, J.E. Hirschman, E.L. Hong, L. Issel-Tarver, R. Nash, A. Sethuraman, B. Starr, C.L. Theesfeld, R. Andrada, G. Binkley, Q. Dong, C. Lane, M. Schroeder, D. Botstein, and J.M. Cherry. Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from *Saccharomyces cerevisiae* and related sequences from other organisms. *Nucleic Acids Res*, 32(Database issue):D311–D314, Jan 2004.

[13] A. Joshi and B.O. Palsson. Metabolic dynamics in the human red cell. Part I–A comprehensive kinetic model. *J Theor Biol*, 141(4):515–528, Dec 1989.

[14] P.J. Mulquiney and P.W. Kuchel. Model of 2,3-bisphosphoglycerate metabolism in the human erythrocyte based on detailed enzyme kinetic equations: computer simulation and metabolic control analysis. *Biochem J*, 342 Pt 3:597–604, Sep 1999.

[15] P.J. Mulquiney and P.W. Kuchel. Model of 2,3-bisphosphoglycerate metabolism in the human erythrocyte based on detailed enzyme kinetic equations: equations and parameter refinement. *Biochem J*, 342 Pt 3:581–596, Sep 1999.

[16] P.J. Mulquiney, W.A. Bubb, and P.W. Kuchel. Model of 2,3-bisphosphoglycerate metabolism in the human erythrocyte based on detailed enzyme kinetic equations: *in vivo* kinetic characterization of 2,3-bisphosphoglycerate synthase/phosphatase using 13C and 31P NMR. *Biochem J*, 342 Pt 3:567–580, Sep 1999.

[17] Y. Nakayama, A. Kinoshita, and M. Tomita. Dynamic simulation of red blood cell metabolism and its application to the analysis of a pathological condition. *Theor Biol Med Model*, 2(1):18, May 2005.

[18] M.W. Covert, C.H. Schilling, I. Famili, J.S. Edwards, I.I. Goryanin, E. Selkov, and B.O. Palsson. Metabolic modeling of microbial strains *in silico*. *Trends Biochem. Sci.*, 26(3):179–186, 2001.

[19] N.D. Price, J.L. Reed, and B.O. Palsson. Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nat Rev Microbiol*, 2(11):886–897, Nov 2004.

[20] J.S. Edwards and B.O. Palsson. The *Escherichia coli* MG1655 *in silico* metabolic genotype: Its definition, characteristics, and capabilities. *PNAS*, 97(10):5528–5533, 2000.

[21] I. Borodina, P. Krabben, and J. Nielsen. Genome-scale analysis of *Streptomyces coelicolor* A3(2) metabolism. *Genome Res.*, 15(6):820–829, 2005.

[22] S.J. Wiback and B.O. Palsson. Extreme pathway analysis of human red blood cell metabolism. *Biophys J*, 83(2):808–818, Aug 2002.

[23] C.H. Schilling, S. Schuster, B.O. Palsson, and R. Heinrich. Metabolic pathway analysis: basic concepts and scientific applications in the post-genomic era. *Biotechnol. Prog.*, 15:296–303, 1999.

[24] C. Reder. Metabolic control theory: a structural approach. *J. Theor. Biol.*, 135:175–201, 1988.

[25] I. Famili and B.O. Palsson. Systemic metabolic reactions are obtained by singular value decomposition of genome-scale stoichiometric matrices. *J Theor Biol*, 224(1):87–96, Sep 2003.

[26] H.M. Sauro and B. Ingalls. Conservation analysis in biochemical networks: computational issues for software writers. *Biophys Chem*, 109(1):1–15, Apr 2004.

[27] J.L. Reed and B.O. Palsson. Thirteen years of building constraint-based *in silico* models of *Escherichia coli*. *J Bacteriol*, 185(9):2692–2699, May 2003.

[28] B. Bollobas. *Graph theory: an introductory course*. Springer-Verlag, 1979.

[29] R. Albert. Scale-free networks in cell biology. *J Cell Sci*, 118(Pt 21):4947–4957, Nov 2005.

[30] A. Wagner and D.A. Fell. The small world inside large metabolic networks. *Proc Biol Sci*, 268(1478):1803–1810, Sep 2001.

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

[32] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, Jun 1998.

[33] D.A. Fell and A. Wagner. The small world of metabolism. *Nat Biotechnol*, 18(11):1121–1122, Nov 2000.

[34] N. Lemke, F. Heredia, C.K. Barcellos, A.N. dos Reis, and J.C.M. Mombach. Essentiality and damage in metabolic networks. *Bioinformatics*, 20(1):115–119, 2004.

[35] I. Famili and B.O. Palsson. The convex basis of the left null space of the stoichiometric matrix leads to the definition of metabolically meaningful pools. *Biophys J*, 85(1):16–26, Jul 2003.

[36] S. Schuster and C. Hilgetag. What information about the conserved-moiety structure of chemical reaction systems can be derived from their stoichiometry? *J. Phys. Chem.*, 99:8017–8023, 1995.

[37] S. Schuster and R. Schuster. Detecting strictly detailed balanced subnetworks in open chemical reaction networks. *Journal of Mathematical Chemistry*, 6(1):17–40, Dec 1991.

[38] A.P. Burgard, E.V. Nikolaev, C.H. Schilling, and C.D. Maranas. Flux coupling analysis of genome-scale metabolic network reconstructions. *Genome Res*, 14(2):301–312, Feb 2004.

[39] T. Pfeiffer, I. Sanchez-Valdenebro, J.C. Nuno, F. Montero, and S. Schuster. METATOOL: for studying metabolic networks. *Bioinformatics*, 15(3):251–257, 1999.

[40] S. Klamt and J. Stelling. Combinatorial complexity of pathway analysis in metabolic networks. *Molecular Biology Reports*, 29(1-2):233–236, 2002.

[41] J.L. Reed and B.O. Palsson. Genome-scale *in silico* models of *E. coli* have multiple equivalent phenotypic states: assessment of correlated reaction subsets that comprise network states. *Genome Res.*, 14(9):1797–1805, 2004.

[42] S. Schuster, S. Klamt, W. Weckwerth, M. Moldenhauer, and T. Pfeiffer. Use of network analysis of metabolic systems in bioengineering. *Bioprocess Biosys. Eng.*, 24:363–373, 2002.

[43] B.K. Bonde. *Metabolism and bioinformatics: the relationship between metabolism and genome structure.* PhD thesis, Oxford Brookes University, 2006.

[44] M.W. Covert, I. Famili, and B.O. Palsson. Identifying constraints that govern cell behavior: a key to converting conceptual to computational models in biology? *Biotechnol Bioeng*, 84(7):763–772, Dec 2003.

[45] N.D. Price, J.A. Papin, C.H. Schilling, and B.O. Palsson. Genome-scale microbial in silico models: the constraints-based approach. *Trends Biotechnol*, 21(4):162–169, Apr 2003.

[46] R.T. Rockafellar. *Convex Analysis.* Princeton Univ. Press, Princeton, 1970.

[47] B.O. Palsson. The challenges of *in silico* biology. *Nat Biotechnol*, 18(11):1147–1150, Nov 2000.

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

[49] S. Schuster and C. Hilgetag. On elementary flux modes in biochemical systems at steady state. *J. Biol. Syst.*, 2:165–182, 1994.

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

[51] G. Stephanopoulos. Metabolic engineering. *Biotechnol Bioeng*, 58(2-3):119–120, 1998.

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

[53] J.A. Papin, N.D. Price, S.J. Wiback, D.A. Fell, and B.O. Palsson. Metabolic pathways in the post-genomic era. *Trends Biochem. Sci.*, 28(5):250–258, 2003.

[54] S. Schuster, D.A. Fell, and T. Dandekar. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotech.*, 18:326–332, 2000.

[55] J. Gagneur and S. Klamt. Computation of elementary modes: a unifying framework and the new binary approach. *BMC Bioinformatics*, 5:175, 2004.

[56] R. Urbanczik and C. Wagner. An improved algorithm for stoichiometric network analysis: theory and applications. *Bioinformatics*, 21(7):1203–1210, Apr 2005.

[57] D.A. Fell. *Understanding the control of metabolism.* Portland Press, London, 1997.

[58] J. Stelling, S. Klamt, B. Bettenbrock, S. Schuster, and E.D. Gilles. Metabolic network structure determines key aspects of functionality and regulation. *Nature*, 420:190–193, 2002.

[59] M.G. Poolman, D.A. Fell, and C.A. Raines. Elementary modes analysis of photosynthate metabolism in the chloroplast stroma. *Eur. J. Biochem.*, 270:430–439, 2003.

[60] R. Patnaik and J.C. Liao. Engineering of *Escherichia coli* central metabolism for aromatic metabolite production with near theoretical yield. *Appl. Environ. Microbiol.*, 60(11):3903–3908, 1994.

[61] R. Carlson, A. Wlaschin, and F. Srienc. Kinetic studies and biochemical pathway analysis of anaerobic poly-(R)-3-hydroxybutyric acid synthesis in *Escherichia coli*. *Appl. Environ. Microbiol.*, 71(2):713–720, 2005.

[62] S. Klamt and J. Stelling. Two approaches for metabolic pathway analysis? *Trends Biotechnol*, 21(2):64–69, Feb 2003.

[63] B.O. Palsson, N.D. Price, and J.A. Papin. Development of network-based pathway definitions: the need to analyze real metabolic networks. *Trends Biotechnol*, 21(5):195–198, May 2003.

[64] N.D. Price, J.A. Papin, and B.O. Palsson. Determination of redundancy and systems properties of the metabolic network of *Helicobacter pylori* using genome-scale extreme pathway analysis. *Genome Res*, 12(5):760–769, May 2002.

[65] R.T.J.M. van der Heijden, B. Romein, J.J. Heijnen, C. Hellinga, and K.Ch.A.M. Luyben. Linear constraint relations in biochemical reaction systems: II. Diagnosis and estimation of gross errors. *Biotechnol. Bioeng.*, 43:11–20, 1994.

[66] S. Klamt, S. Schuster, and E.D. Gilles. Calculability analysis in underdetermined metabolic networks illustrated by a model of the central metabolism in purple nonsulfur bacteria. *Biotechnol Bioeng*, 77(7):734–751, Mar 2002.

[67] W. Wiechert. 13C metabolic flux analysis. *Metab Eng*, 3(3):195–206, Jul 2001.

[68] D. Segre, D. Vitkup, and G.M. Church. Analysis of optimality in natural and perturbed metabolic networks. *Proc Natl Acad Sci U S A*, 99(23):15112–15117, Nov 2002.

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

[70] S. Lee, C. Palakornkule, M.M. Domach, and I.E. Grossmann. Recursive MILP model for finding all the alternate optima in LP models for metabolic networks. *Computers and Chemical Engineering*, 24(2):711–716, July 2000.

[71] C. Phalakornkule, S. Lee, T. Zhu, R. Koepsel, M.M. Ataai, I.E. Grossmann, and M.M. Domach. A MILP-based flux alternative generation and NMR experimental design strategy for metabolic engineering. *Metab Eng*, 3(2):124–137, Apr 2001.

[72] R. Mahadevan and C.H. Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metab Eng*, 5(4):264–276, Oct 2003.

[73] J. Forster, I. Famili, P. Fu, B. Palsson, and J. Nielsen. Genome-scale reconstruction of the *Saccharomyces cerevisiae* metabolic network. *Genome Res.*, 13(2):244–253, 2003.

[74] J.S. Edwards and B.O. Palsson. Systems properties of the *Haemophilus influenzae* Rd metabolic genotype. *J Biol Chem*, 274(25):17410–17416, Jun 1999.

[75] J.S. Edwards and B.O. Palsson. Robustness analysis of the *Escherichia coli* metabolic network. *Biotechnol Prog*, 16(6):927–939, 2000.

[76] S. Becker and B.O. Palsson. Genome-scale reconstruction of the metabolic network in *Staphylococcus aureus* N315: an initial draft to the two-dimensional annotation. *BMC Microbiology*, 5(1):8, 2005.

[77] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. COPASI–a COmplex PAthway SImulator. *Bioinformatics*, 22(24):3067–3074, Dec 2006.

[78] P. Mendes. GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Comput Appl Biosci*, 9(5):563–571, Oct 1993.

[79] H.M. Sauro. Jarnac: a system for interactive metabolic analysis. In *Hofmeyr, J.H., Rohwer, J.M., Snoep, J.L. (Eds.). Animating the Cellular Map: Proceedings of the 9th International Meeting on BioThermoKinetics*, pages 221–228. Stellenbosch University Press, 2000.

[80] H.M. Sauro. SCAMP: a general-purpose simulator and metabolic control analysis program. *Comput Appl Biosci*, 9(4):441–450, Aug 1993.

[81] B.G. Olivier, J.M. Rohwer, and J.S. Hofmeyr. Modelling cellular systems with PySCeS. *Bioinformatics*, 21(4):560–561, Feb 2005.

[82] M. Lutz and D. Ascher. *Learning Python*. O'Reilly & Associates, 1999.

[83] R. Schwarz, P. Musch, A. von Kamp, B. Engels, H. Schirmer, S. Schuster, and T. Dandekar. YANA - a software tool for analyzing flux modes, gene-expression and enzyme activities. *BMC Bioinformatics*, 6:135, 2005.

[84] S. Klamt, J. Saez-Rodriguez, and E.D. Gilles. Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Syst Biol*, 1:2, 2007.

[85] S. Klamt, J. Stelling, M. Ginkel, and E.D. Gilles. FluxAnalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. *Bioinformatics*, 19(2):261–269, Jan 2003.

[86] M.G. Poolman. ScrumPy: metabolic modelling with Python. *IEE Proc Syst Biol*, 153(5):375–378, Sep 2006.

[87] L. Prechelt. An empirical comparison of seven programming languages. *Computer*, 33(10):23–29, 2000.

[88] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25):14863–14868, 1998.

[89] D. Hand, H. Mannila, and P. Smyth. *Principles of data mining*. MIT Press, Cambridge, MA, 2001.

[90] E. Alpaydin. *Introduction to machine learning*. MIT Press, Cambridge, MA, 2004.

[91] C.M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[92] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[93] M.N. Murty and A.K. Jain. Knowledge-base clustering scheme for collection management and retrieval of library books. *Pattern Recognition*, 28(7):949–963, July 1995.

[94] Chris Mueller website. `http://www.osl.iu.edu/~chemuell/new/index.php`.

[95] J.B. MacQueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[96] F.D. Smet, J. Mathys, K. Marchal, G. Thijs, B. De Moor, and Y. Moreau. Adaptive quality-based clustering of gene expression profiles. *Bioinformatics*, 18(5):735–746, May 2002.

[97] A.K. Jain and R.C. Dubes. *Algorithms for clustering data.* Prentice-Hall, Inc., 1988.

[98] J. Mao and A.K. Jain. A self-organizing network for hyperellipsoidal clustering (HEC). *IEEE Trans. Neural Netw.*, 7:16–29, 1996.

[99] Gene Expression Pattern Analysis Suite. `http://bioinfo.cipf.es/wikigepas/clustering`.

[100] L. Kaufman and P.J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis.* Wiley and Sons, 1990.

[101] G.W. Milligan. An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika*, 45:325–342, 1980.

[102] D.P. Labeda. Transfer of the type strain of *Streptomyces erythraeus* (Waksman 1923) Waksman and Henrici 1948 to the genus *Saccharopolyspora* Lacey and Goodfellow 1975 as *Saccharopolyspora erythraea* sp. nov., and designation of a neotype strain for *Streptomyces erythraeus. Int. J. Syst. Bacteriol.*, 37(1):19–22, Jan 1987.

[103] J. Staunton and K.J. Weissman. Polyketide biosynthesis: a millennium review. *Nat Prod Rep*, 18(4):380–416, Aug 2001.

[104] H. Kitano. Computational systems biology. *Nature*, 420(6912):206–210, Nov 2002.

[105] C. Francke, R.J. Siezen, and B. Teusink. Reconstructing the metabolic network of a bacterium from its genome. *Trends Microbiol*, 13(11):550–558, Nov 2005.

[106] C.H. Schilling, M.W. Covert, I. Famili, G.M. Church, J.S. Edwards, and B.O. Palsson. Genome-scale metabolic model of *Helicobacter pylori* 26695. *J. Bacteriol.*, 184(16):4582–4593, 2002.

[107] M. Kanehisa, S. Goto, M. Hattori, K.F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res*, 34:D354–D357, Jan 2006.

[108] U. Wittig and A. De Beuckelaer. Analysis and comparison of metabolic pathway databases. *Brief Bioinform*, 2(2):126–142, May 2001.

[109] K.D. Pruitt, T. Tatusova, and D.R. Maglott. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res*, 35(Database issue):D61–D65, Jan 2007.

[110] S. Goto, Y. Okuno, M. Hattori, T. Nishioka, and M. Kanehisa. LIGAND: database of chemical compounds and reactions in biological pathways. *Nucleic Acids Res*, 30(1):402–404, Jan 2002.

[111] W. Fujibuchi, S. Goto, H. Migimatsu, I. Uchiyama, A. Ogiwara, Y. Akiyama, and M. Kanehisa. DBGET/LinkDB: an integrated database retrieval system. *Pac Symp Biocomput*, pages 683–694, 1998.

[112] P.D. Karp, S. Paley, and P. Romero. The Pathway Tools software. *Bioinformatics*, 18 Suppl 1:S225–S232, 2002.

[113] B.O. Palsson. Two-dimensional annotation of genomes. *Nat Biotechnol*, 22(10):1218–1219, Oct 2004.

[114] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, Oct 1990.

[115] W.R. Pearson and D.J. Lipman. Improved tools for biological sequence comparison. *Proc Natl Acad Sci USA*, 85(8):2444–2448, Apr 1988.

[116] E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R.D. Appel, and A. Bairoch. ExPASy: The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res*, 31(13):3784–3788, Jul 2003.

[117] C. Claudel-Renard, T. Chevalet, C. andFaraut, and D. Kahn. Enzyme-specific profiles for genome annotation: PRIAM. *Nucleic Acids Res*, 31(22):6633–6639, Nov 2003.

[118] J.W. Pinney, M.W. Shirley, G.A. McConkey, and D.R. Westhead. metaSHARK: software for automated metabolic network prediction from DNA sequence and its application to the genomes of *Plasmodium falciparum* and *Eimeria tenella*. *Nucleic Acids Res.*, 33(4):1399–1409, 2005.

[119] N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, B.A. Cuche, E. Castro, C. Lachaize, P.S. Langendijk-Genevaux, and C.J.A. Sigrist. The 20 years of PROSITE. *Nucleic Acids Res*, 36(Database issue):D245–D249, Jan 2008.

[120] C.J.A. Sigrist, L. Cerutti, N. Hulo, A. Gattiker, L. Falquet, M. Pagni, A. Bairoch, and P. Bucher. PROSITE: a documented database using patterns and profiles as motif descriptors. *Brief Bioinform*, 3(3):265–274, Sep 2002.

[121] N.J. Mulder, R. Apweiler, T.K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bork, V. Buillard, L. Cerutti, R. Copley, E. Courcelle, U. Das, L. Daugherty, M. Dibley, R. Finn, W. Fleischmann, J. Gough, D. Haft, N. Hulo, S. Hunter, D. Kahn, A. Kanapin, A. Kejariwal, A. Labarga, P.S. Langendijk-Genevaux, D. Lonsdale, R. Lopez, I. Letunic, M. Madera, J. Maslen, C. McAnulla, J. McDowall, J. Mistry, A. Mitchell, A.N. Nikolskaya, S. Orchard, C. Orengo, R. Petryszak, J.D. Selengut, C.J.A. Sigrist, P.D. Thomas, F. Valentin, D. Wilson, C.H. Wu, and C. Yeats. New developments in the InterPro database. *Nucleic Acids Res*, 35(Database issue):D224–D228, Jan 2007.

[122] A. Bairoch. The ENZYME database in 2000. *Nucleic Acids Res*, 28(1):304–305, Jan 2000.

[123] J. Gouzy, F. Corpet, and D. Kahn. Whole genome protein domain analysis using a new method for domain clustering. *Comput Chem*, 23(3-4):333–340, Jun 1999.

[124] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402, Sep 1997.

[125] A. Marchler-Bauer, J.B. Anderson, C. DeWeese-Scott, N.D. Fedorova, L.Y. Geer, S. He, D.I. Hurwitz, J.D. Jackson, A.R. Jacobs, C.J. Lanczycki, C.A.

Liebert, C. Liu, T. Madej, G.H. Marchler, R. Mazumder, A.N. Nikolskaya, A.R. Panchenko, B.S. Rao, B.A. Shoemaker, V. Simonyan, J.S. Song, P.A. Thiessen, S. Vasudevan, Y. Wang, R.A. Yamashita, J.J. Yin, and S.H. Bryant. CDD: a curated Entrez database of conserved domain alignments. *Nucleic Acids Res*, 31(1):383–387, Jan 2003.

[126] J.L. Reed, I. Famili, I. Thiele, and B.O. Palsson. Towards multidimensional genome annotation. *Nat Rev Genet*, 7(2):130–141, Feb 2006.

[127] M. Riley, T. Abe, M.B. Arnaud, M.K.B. Berlyn, F.B. Blattner, R.R. Chaudhuri, J.D. Glasner, T. Horiuchi, I.M. Keseler, T. Kosuge, H. Mori, N.T. Perna, G. Plunkett, K.E. Rudd, M.H. Serres, G.H. Thomas, N.R. Thomson, D. Wishart, and B.L. Wanner. *Escherichia coli* K-12: a cooperatively developed annotation snapshot–2005. *Nucleic Acids Res*, 34(1):1–9, 2006.

[128] R.A. Notebaart, F.H.J. van Enckevort, C. Francke, R.J. Siezen, and B. Teusink. Accelerating the reconstruction of genome-scale metabolic networks. *BMC Bioinformatics*, 7:296, 2006.

[129] M. DeJongh, K. Formsma, P. Boillot, J. Gould, M. Rycenga, and A. Best. Toward the automated generation of genome-scale metabolic networks in the SEED. *BMC Bioinformatics*, 8:139, 2007.

[130] A. Osterman and R. Overbeek. Missing genes in metabolic pathways: a comparative genomics approach. *Curr Opin Chem Biol*, 7(2):238–251, Apr 2003.

[131] J.S. Edwards and B.O. Palsson. How will bioinformatics influence metabolic engineering? *Biotechnol Bioeng*, 58(2-3):162–169, 1998.

[132] J.A. Papin, N.D. Price, J.S. Edwards, and B.O. Palsson. The genome-scale metabolic extreme pathway structure in *Haemophilus influenzae* shows significant network redundancy. *J. Theor. Biol.*, 215:67–82, 2002.

[133] N.C. Duarte, M.J. Herrgard, and B.O. Palsson. Reconstruction and validation of *Saccharomyces cerevisiae* iND750, a fully compartmentalized genome-scale metabolic model. *Genome Res*, 14(7):1298–1309, Jul 2004.

[134] M.G. Poolman, B.K. Bonde, A. Gevorgyan, H.H. Patel, and D.A. Fell. Challenges to be faced in the reconstruction of metabolic networks from public databases. *IEE Proc. Systems Biol.*, 153(5):379–384, September 2006.

[135] P.M. Hammond. *Described and estimated species numbers: an objective assessment of current knowledge*, chapter Microbial diversity and ecosystem function., pages 29–71. CAB International; Wallingford, UK, 1995.

[136] G.J. Olsen, C.R. Woese, and R. Overbeek. The winds of (evolutionary) change: breathing new life into microbiology. *J Bacteriol*, 176(1):1–6, Jan 1994.

[137] R.C. Edgar and S. Batzoglou. Multiple sequence alignment. *Curr Opin Struct Biol*, 16(3):368–373, Jun 2006.

[138] C.R. Woese and G.E. Fox. Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proc Natl Acad Sci U S A*, 74(11):5088–5090, Nov 1977.

[139] G.E. Fox, K.R. Pechman, and C.R. Woese. Comparative cataloging of 16S ribosomal ribonucleic acid: molecular approach to procaryotic systematics. *Int J Syst Bacteriol*, 27(1):44–57, 1977.

[140] C.R. Woese. Bacterial evolution. *Microbiol Rev*, 51(2):221–271, Jun 1987.

[141] C.R. Woese, O. Kandler, and M.L. Wheelis. Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya. *Proc Natl Acad Sci U S A*, 87(12):4576–4579, Jun 1990.

[142] J.R. Brown and W.F. Doolittle. Archaea and the prokaryote-to-eukaryote transition. *Microbiol Mol Biol Rev*, 61(4):456–502, Dec 1997.

[143] W.H. Yap, Z. Zhang, and Y. Wang. Distinct types of rRNA operons exist in the genome of the actinomycete *Thermomonospora chromogena* and evidence for horizontal transfer of an entire rRNA operon. *J Bacteriol*, 181(17):5201–5209, Sep 1999.

[144] H. Philippe and J. Laurent. How good are deep phylogenetic trees? *Curr Opin Genet Dev*, 8(6):616–623, Dec 1998.

[145] S.R. Henz, D.H. Huson, A.F. Auch, K. Nieselt-Struwe, and S.C. Schuster. Whole-genome prokaryotic phylogeny. *Bioinformatics*, 21(10):2329–2335, May 2005.

[146] D.D. Leipe, Y.I. Wolf, E.V. Koonin, and L. Aravind. Classification and evolution of P-loop GTPases and related ATPases. *J Mol Biol*, 317(1):41–72, Mar 2002.

[147] W.F. Doolittle. Phylogenetic classification and the universal tree. *Science*, 284(5423):2124–2129, Jun 1999.

[148] Y.I. Wolf, I.B. Rogozin, N.V. Grishin, R.L. Tatusov, and E.V. Koonin. Genome trees constructed using five different approaches suggest new major bacterial clades. *BMC Evol Biol*, 1:8, Oct 2001.

[149] G.D.P. Clarke, R.G. Beiko, M.A. Ragan, and R.L. Charlebois. Inferring genome trees by using a filter to eliminate phylogenetically discordant sequences and a distance matrix based on mean normalized BLASTP scores. *J Bacteriol*, 184(8):2072–2080, Apr 2002.

[150] A.K. Bansal and T.E. Meyer. Evolutionary analysis by whole-genome comparisons. *J Bacteriol*, 184(8):2260–2272, Apr 2002.

[151] V. Daubin, M. Gouy, and G. Perriere. Bacterial molecular phylogeny using supertree approach. *Genome Inform*, 12:155–164, 2001.

[152] M.C. Rivera, Jain R., J.E. Moore, and J.A. Lake. Genomic evidence for two functionally distinct gene classes. *Proc Natl Acad Sci U S A*, 95(11):6239–6244, May 1998.

[153] E.V. Koonin, A.R. Mushegian, M.Y. Galperin, and D.R. Walker. Comparison of archaeal and bacterial genomes: computer analysis of protein sequences predicts novel functions and suggests a chimeric origin for the archaea. *Mol Microbiol*, 25(4):619–637, Aug 1997.

[154] E.V. Koonin, K.S. Makarova, and L. Aravind. Horizontal gene transfer in prokaryotes: quantification and classification. *Annu Rev Microbiol*, 55:709–742, 2001.

[155] E. Hilario and J.P. Gogarten. Horizontal transfer of ATPase genes–the tree of life becomes a net of life. *Biosystems*, 31(2-3):111–119, 1993.

[156] S.T. Fitz-Gibbon and C.H. House. Whole genome-based phylogenetic analysis of free-living microorganisms. *Nucleic Acids Res*, 27(21):4218–4222, Nov 1999.

[157] B. Snel, P. Bork, and M.A. Huynen. Genome phylogeny based on gene content. *Nat Genet*, 21(1):108–110, Jan 1999.

[158] J.O. Korbel, B. Snel, B.A. Huynen, and P. Bork. SHOT: a web server for the construction of genome phylogenies. *Trends Genet*, 18(3):158–162, Mar 2002.

[159] V. Kunin, D. Ahren, L. Goldovsky, P. Janssen, and C.A. Ouzounis. Measuring genome conservation across taxa: divided strains and united kingdoms. *Nucleic Acids Res*, 33(2):616–621, 2005.

[160] T. Coenye and P. Vandamme. Extracting phylogenetic information from whole-genome sequencing projects: the lactic acid bacteria as a test case. *Microbiology*, 149(Pt 12):3507–3517, Dec 2003.

[161] Y. Zhang, S. Li, G. Skogerbo, Z. Zhang, X. Zhu, Z. Zhang, S. Sun, H. Lu, B. Shi, and R. Chen. Phylophenetic properties of metabolic pathway topologies as revealed by global analysis. *BMC Bioinformatics*, 7:252, 2006.

[162] D. Aguilar, F.X. Aviles, E. Querol, and M.J.E. Sternberg. Analysis of phenetic trees based on metabolic capabilites across the three domains of life. *J Mol Biol*, 340(3):491–512, Jul 2004.

[163] C.V. Forst and K. Schulten. Evolution of metabolisms: a new method for the comparison of metabolic pathways using genomics information. *J Comput Biol*, 6(3-4):343–360, 1999.

[164] C.V. Forst and K. Schulten. Phylogenetic analysis of metabolic pathways. *J Mol Evol*, 52(6):471–489, Jun 2001.

[165] M. Heymans and A.K. Singh. Deriving phylogenetic trees from the similarity analysis of metabolic pathways. *Bioinformatics*, 19 Suppl 1:i138–i146, 2003.

[166] L. Liao, S. Kim, and J. Tomb. Genome comparisons based on profiles of metabolic pathways. In *Proceedings of the 6th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, pages 469–476, 2002.

[167] H. Ma and A. Zeng. Phylogenetic comparison of metabolic capacities of organisms at genome level. *Mol Phylogenet Evol*, 31(1):204–213, Apr 2004.

[168] J. Podani, Z.N. Oltvai, H. Jeong, B. Tombor, A.L. Barabasi, and E. Szathmary. Comparable system-level organization of Archaea and Eukaryotes. *Nat Genet*, 29(1):54–56, Sep 2001.

[169] S.H. Hong, T.Y. Kim, and S.Y. Lee. Phylogenetic analysis based on genome-scale metabolic pathway reaction content. *Appl Microbiol Biotechnol*, 65(2):203–210, Aug 2004.

[170] R.L. Tatusov, D.A. Natale, I.V. Garkavtsev, T.A. Tatusova, U.T. Shankavaram, B.S. Rao, B. Kiryutin, M.Y. Galperin, N.D. Fedorova, and

E.V. Koonin. The COG database: new developments in phylogenetic classification of proteins from complete genomes. *Nucleic Acids Res*, 29(1):22–28, Jan 2001.

[171] D. Zhu and Z.S. Qin. Structural comparison of metabolic networks in selected single cell organisms. *BMC Bioinformatics*, 6:8, 2005.

[172] W. Liu, W. Lin, A.J. Davis, F. Jordan, H. Yang, and M. Hwang. A network perspective on the topological importance of enzymes and their phylogenetic conservation. *BMC Bioinformatics*, 8:121, 2007.

[173] C.V. Forst, C. Flamm, I.L. Hofacker, and P.F. Stadler. Algebraic comparison of metabolic networks, phylogenetic inference, and metabolic innovation. *BMC Bioinformatics*, 7:67, 2006.

[174] S.G. Acinas, L.A. Marcelino, V. Klepac-Ceraj, and M.F. Polz. Divergence and redundancy of 16S rRNA sequences in genomes with multiple *rrn* operons. *J Bacteriol*, 186(9):2629–2635, May 2004.

[175] A.J. Mackey, T.A.J. Haystead, and W.R. Pearson. Getting more from less: algorithms for rapid protein identification with multiple short peptide sequences. *Mol Cell Proteomics*, 1(2):139–147, Feb 2002.

[176] M.A. Larkin, G. Blackshields, N.P. Brown, R. Chenna, P.A. McGettigan, H. McWilliam, F. Valentin, I.M. Wallace, A. Wilm, R. Lopez, J.D. Thompson, T.J. Gibson, and D.G. Higgins. Clustal W and Clustal X version 2.0. *Bioinformatics*, 23(21):2947–2948, Nov 2007.

[177] C. Notredame, D.G. Higgins, and J. Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol*, 302(1):205–217, Sep 2000.

[178] R.C. Edgar. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5:113, Aug 2004.

[179] D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2):131–147, February 1981.

[180] Y.I. Wolf, I.B. Rogozin, N.V. Grishin, and E.V. Koonin. Genome trees and the tree of life. *Trends Genet*, 18(9):472–479, Sep 2002.

[181] P. Sirand-Pugnet, C. Lartigue, M. Marenda, D. Jacob, A. Barre, V. Barbe, C. Schenowitz, S. Mangenot, A. Couloux, B. Segurens, A. de Daruvar, A. Blanchard, and C. Citti. Being pathogenic, plastic, and sexual while

living with a nearly minimal bacterial genome. *PLoS Genet*, 3(5):e75, May 2007.

[182] R.D. Fleischmann, M.D. Adams, O. White, R.A. Clayton, E.F. Kirkness, A.R. Kerlavage, C.J. Bult, J.F. Tomb, B.A. Dougherty, and J.M. Merrick. Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science*, 269(5223):496–512, Jul 1995.

[183] R. Overbeek, T. Begley, R.M. Butler, J.V. Choudhuri, H. Chuang, M. Cohoon, V. Crecy-Lagard, N. Diaz, T. Disz, R. Edwards, M. Fonstein, E.D. Frank, S. Gerdes, E.M. Glass, A. Goesmann, A. Hanson, D. Iwata-Reuyl, R. Jensen, N. Jamshidi, L. Krause, M. Kubal, N. Larsen, B. Linke, A.C. McHardy, F. Meyer, H. Neuweger, G. Olsen, R. Olson, A. Osterman, V. Portnoy, G.D. Pusch, D.A. Rodionov, C. Rckert, J. Steiner, R. Stevens, I. Thiele, O. Vassieva, Y. Ye, O. Zagnitko, and V. Vonstein. The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. *Nucleic Acids Res*, 33(17):5691–5702, 2005.

[184] K.B. Cohen and L. Hunter. Getting started in text mining. *PLoS Comput Biol*, 4(1):e20, Jan 2008.

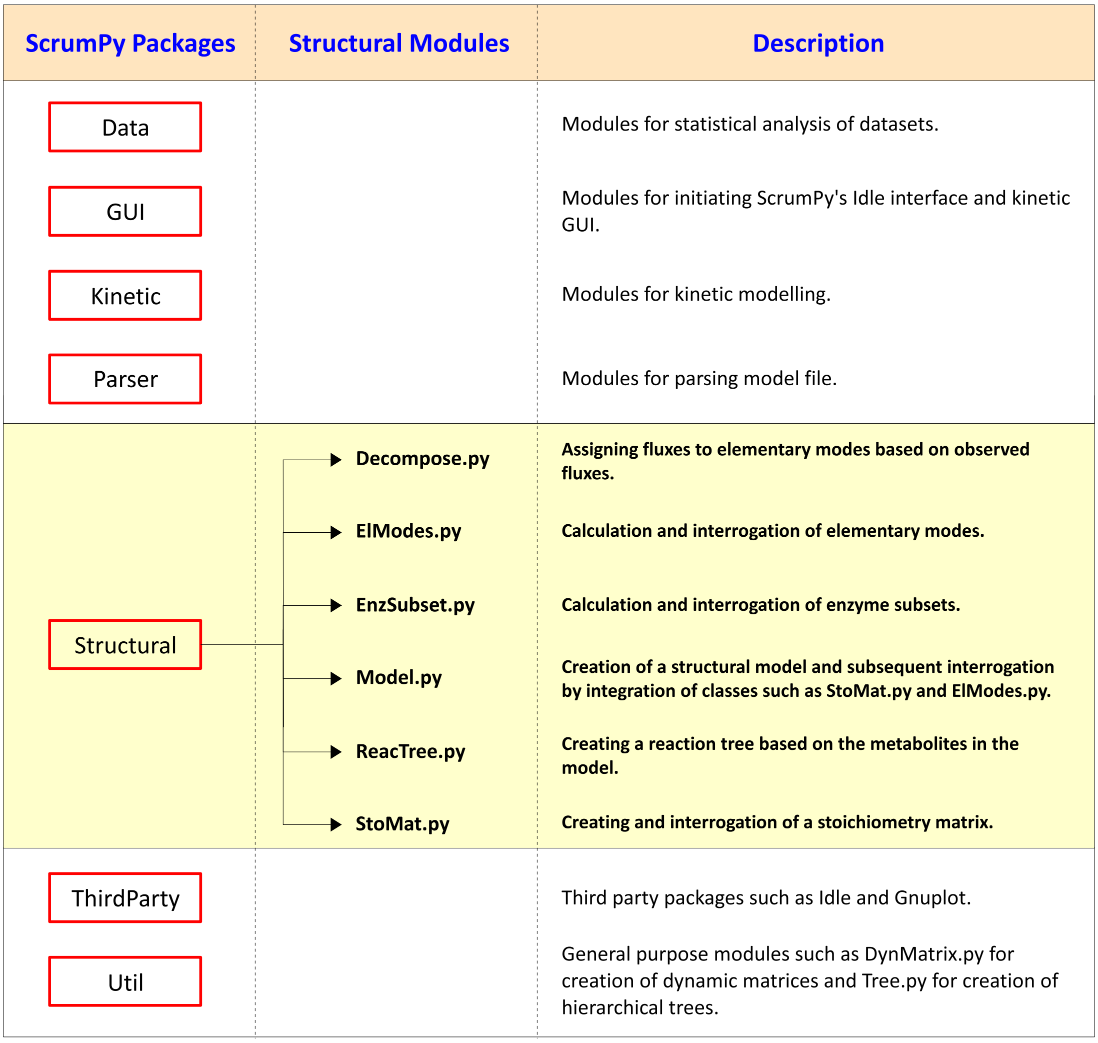
[185] M.G. Poolman, K.V. Venkatesh, M.K. Pidcock, and D.A. Fell. A method for the determination of flux in elementary modes, and its application to *Lactobacillus rhamnosus*. *Biotechnol Bioeng*, 88(5):601–612, Dec 2004.

APPENDIX A

Using ScrumPy: a Basic Guide

A.1 ScrumPy packages

Although ScrumPy can be used for kinetic modelling, this guide will focus primarily on the structural modelling aspects of the software that have been used throughout the course of this project. The primary packages in ScrumPy are shown in Figure A.1, or alternatively the package structure can be browsed from the ScrumPy menu (i.e. `File/PathBrowser/ScrumPy/`).



ScrumPy Packages	Structural Modules	Description
Data		Modules for statistical analysis of datasets.
GUI		Modules for initiating ScrumPy's Idle interface and kinetic GUI.
Kinetic		Modules for kinetic modelling.
Parser		Modules for parsing model file.
Structural	**Decompose.py**	**Assigning fluxes to elementary modes based on observed fluxes.**
	ElModes.py	**Calculation and interrogation of elementary modes.**
	EnzSubset.py	**Calculation and interrogation of enzyme subsets.**
	Model.py	**Creation of a structural model and subsequent interrogation by integration of classes such as StoMat.py and ElModes.py.**
	ReacTree.py	**Creating a reaction tree based on the metabolites in the model.**
	StoMat.py	**Creating and interrogation of a stoichiometry matrix.**
ThirdParty		Third party packages such as Idle and Gnuplot.
Util		General purpose modules such as DynMatrix.py for creation of dynamic matrices and Tree.py for creation of hierarchical trees.

Figure A.1 – Important ScrumPy packages, with particular emphasis on the Structural package, along with brief descriptions. See [185] for background and application of contents in Decompose.py module.

A.2 Model definition

The model file must be converted into ScrumPy's native '*.spy*' format (Figure A.2) before it can be loaded. The minimal information required to reconstruct a structural model is a list of reaction names along with their associated stoichiometric equations. To avoid errors when parsing the model file certain rules apply when defining these items:

- Reaction names must begin with an upper or lower case letter, followed by any sequence of letters, digits or "_" (underscore). Other characters such as white space are not allowed (see point 5). Additionally, a ":" (colon) delimiter must be placed after all reaction names.

- Metabolite names are defined in the same way as reaction names, except that they cannot contain a colon anywhere. Metabolites prefixed by "`x_`" or "`X_`" are defined as external and the remainder are considered to be internal (free or floating).

- The reaction symbol for a reversible reaction is "`<>`" and "`->`" for an irreversible reaction.

- A stoichiometric equation must consist of substrate names, a reaction symbol and product names. More than one substrate or product is separated by a "+" symbol and any metabolite with a stoichiometric coefficient greater than one is indicated with a positive integer followed by whitespace (e.g. `PPi -> 2 Pi`).

- Using the " (i.e. double quote) character on either side of a reaction or metabolite name allows the modeller to use customised names with characters that would not usually be permitted in the ScrumPy model definition (e.g. white spaces and dashes).

- When carrying out kinetic modelling, the kinetic functions for each reaction also have to be specified. Default rate equations are assigned to the reactions in a structural model by the ~ suffix after the stoichiometric equation for each reaction.

Furthermore, text proceeding "`#`" is ignored by ScrumPy's model parser and serves to provide supplementary model information or notes. ScrumPy directives[1] are optional but useful syntax declarations that exist to treat the model description in a particular manner. They can be defined at the top of the model file and include:

- `Structural()`
  is very useful to ignore the model definition for kinetic information and, consequently, ScrumPy treats the model as purely structural.

[1] can be likened to Python methods (i.e. method name, followed by a possible empty parenthesised parameter list) but have no return value.

- `External(mets)`
  where `mets` is a comma delimited list of metabolites that are to be defined as external to the system (e.g. `External(Water,Oxygen)`). All the metabolites in this list are added to those that have already been prefixed with "`x_`" or "`X_`" to generate the final list of external metabolites.

- `AutoExtern()`
  along with all metabolites prefixed with "`x_`" or "`X_`" and those defined using the `External()` directive, all orphan metabolites (Section 1.6.1.2) are automatically made external.

- `ElType(type)`
  specifies the data type to be used for the elements in the stoichiometry matrices for the model, once it is loaded. `type` can be `int`, `float` or `ArbRat` (default). `ArbRat` or arbitrary-precision rational numbers are used in ScrumPy for exact rational arithmetic and are unsusceptible to round-off or overflow errors. For larger models, `ArbRat` can be replaced with `int` and `float` to increase performance but at the expense of other functionality.

- `Include(models)`
  can be used to collate a number of independent models into a single one. `models` may be a single model filename path or a comma delimited list. Alternatively, more than one `Include()` may be utilised. Extra precaution must be taken to make sure that all reaction names are unique and a global `External()` directive is used between models.

A.3 Loading a model

Once the user is content with the '*.spy*' model file it can be loaded into ScrumPy by typing:

```
>>> model = ScrumPy.Model()
```

at the interactive prompt, where `model` is an instance of a `ScrumPy.Model` object that will be created by parsing the model file. Subsequent model interrogation can be performed by querying and manipulation of such objects. An "`Open File`" dialogue will appear asking for the destination path of a ScrumPy model file. Once selected an editor window will automatically open up to view the model description. Alternatively, a new model can be loaded by passing in a filename argument:

```
############################################################
## Test structural model for the upper half of glycolysis  #
## 18/10/07                                                #
############################################################

   Structural()

   R1:
      X_GLC <> GLC ~
   R2:
      X_ATP + GLC -> G6P + X_ADP ~
   R3:
      G6P <> F6P ~
   R4:
      F6P + X_ATP <> FBP + X_ADP ~
   R5:
      FBP <> DHAP + GAP ~
   R6:
      GAP <> DHAP ~
   R7:
      DHAP <> X_DHAP ~
   R8:
      GAP <> X_GAP ~
   R9:
      G6P <> G1P ~

############################################################
```

Figure A.2 – Format of a ScrumPy (‘*.spy*’) plain ASCII file for the simple metabolic network in Figure 1.4.

```
>>> model = ScrumPy.Model(‘testmodel.spy’)
```

with the ‘*.spy*’ extension. Errors in the input file are displayed in an error message window and highlighted in the model editor window. These errors can also be viewed in the ScrumPy-IDLE interface. If the model description is changed, it may be recompiled by:

```
>>> model.Reload()
```

or via the model editor window by clicking on `ScrumPy/Compile`. Multiple models may be loaded simultaneously but it would not be recommended to load the same model twice.

A.4 Model interrogation and analyses

It is worth reemphasising here that the primary advantage of ScrumPy is the Python programming component which can be exploited to make model interrogation and analyses more flexible. The bulk of modelling functionality in ScrumPy is already provided by using Python *method* attributes:

```
>>> model.Method()
```

where dot notation is used to access the attributes for a Python object, `Method` identifies the attribute and the parentheses indicate that the method is to be invoked. After loading a model into ScrumPy, there are numerous existing analytical methods that are already attached to the `model` object. As with any other Python object, typing `dir(model)` at the prompt returns a list of attributes now comprised within the new `ScrumPy.Model` instance.

A.4.1 Dynamic matrix

The results from most structural analyses carried out using ScrumPy are returned in the form of dynamic matrices (`ScrumPy/Utils/DynMatrix`). A dynamic matrix object is a Python representation of a mathematical matrix and can be used independently from ScrumPy, although it does depend on some of the other modules in the `ScrumPy/Utils` directory. The first step in the creation of a dynamic matrix object is to import the `DynMatrix` module:

```
>>> from ScrumPy.Utils import DynMatrix
```

An instance of a dynamic matrix can be created by using the `matrix` class in `DynMatrix`:

```
>>> mtx = DynMatrix.matrix(nrows, ncols, Conv)
```

where the arguments `nrows` and `ncols` are the numbers of rows and columns respectively, and `Conv` is the required data type for the elements in the matrix (e.g. `int`, `float` and the default `ArbRat`). An empty matrix without any rows and columns, and with arbitrary rational element types will be created if none of these arguments are specified. New instances of a dynamic matrix can also be created from existing matrices:

```
>>> newmtx = mtx.Copy(float)
```

where the optional data type conversion argument has been set to `float`. Depending on the analysis in question there are a variety of interrogation methods attached to these matrices (Figure A.3).

Figure A.3 – Code to illustrate how ScrumPy can be used in conjuction with Python to create and access entries in a dynamic matrix.

```
>>>
>>> mtx = DynMatrix.matrix(Conv=int)   ##create empty matrix object
>>>
>>> rownames = [‘met1’,‘met2’,‘met3’]  ##list of row names
>>> colnames = [‘rxn1’,‘rxn2’,‘rxn3’]  ##list of column names
>>>
>>> for rn in rownames:                ##for all rownames
      mtx.NewRow(name=rn)              ##create a new row and
>>>                                    ##assign it rn
>>> for cn in colnames:                ##repeat for columns
      mtx.NewCol(name=cn)
>>>
>>> mtx                                ##prints mtx to screen
        ‘rxn1’, ‘rxn2’, ‘rxn3’
‘met1’ [  0,      0,      0   ]
‘met2’ [  0,      0,      0   ]
‘met3’ [  0,      0,      0   ]
>>>
>>> mtx.rnames                         ##list of row names
[‘met1’,‘met2’,‘met3’]
>>> mtx.cnames                         ##repeat for columns
[‘rxn1’,‘rxn2’,‘rxn3’]
>>>
>>> mtx[‘met1’,‘rxn2’] = 5      ##assign a value of 5 to element
>>> mtx[1,2] = 7                ##alternative to mtx[rowname,colname]
>>>                             ##but by index
>>>
>>> mtx                         ##print contents of mtx
        ‘rxn1’, ‘rxn2’, ‘rxn3’
‘met1’ [  0,      5,      0   ]
‘met2’ [  0,      0,      7   ]
‘met3’ [  0,      0,      0   ]
>>>
>>> mtx[1]                     ##a single index returns a row
[0, 0, 7]
>>> mtx[‘met2’]                ##or use name to returns a row
[0, 0, 7]
>>>
>>> mtx.GetCol(2)              ##gets column with index 2
[5, 0, 0]                      ##as a list of numbers
>>> mtx.GetRow(‘met2’)         ##gets row with name ‘met2’
[0, 0, 7]
>>>
```

A.4.2 Stoichiometry matrix

A stoichiometry matrix is automatically calculated for `model` once it is loaded into ScrumPy. It can be safely accessed by making a copy:

```
>>> stomat = model.sm.Copy()
```

where the `stomat` instance is as defined in the `ScrumPy.Structural.StoMat` module and contains all the attributes of a dynamic matrix (i.e. subclass). There are a wide-range of ScrumPy methods by which the stoichiometry matrix can be manipulated to fit the needs of the modeller. Herein, we will assume that a `model` object has been loaded from the specification in Figure A.2 and will be used to illustrate the results for the remainder of structural analyses. For example, finding the connectivity of `FBP` in `model` is as simple as typing:

```
>>> stomat.Connectedness('FBP')
2
```

where `Connectedness` is a method of the `model.sm` class that is used to retrieve the stoichiometric data for `FBP` from `stomat`. Using this method it is then possible to find all metabolites in Figure A.2 above a connectivity of two with four lines of Python code:

```
>>> for met in stomat.rnames:
        connectivity = stomat.Connectedness(met)
        if connectivity > 2:
                print connectivity,
                print met
3 G6P
3 DHAP
3 GAP
```

where `rnames` is a dynamic matrix attribute which returns a list of all the metabolite names in the stoichiometry matrix. Orphan metabolites can be obtained from the `stomat` object by:

```
>>> orp = stomat.OrphanMets()
>>> orp
['G1P']
```

where `orp` is a Python list of orphan metabolite names.

A.4.3 Conserved moieties

Conservation relationships within the model definition can be determined by:

```
>>> conmo = model.ConsMoieties()
```

where `conmo` is a ScrumPy dynamic matrix with the metabolites involved with conserved moieties in rows, the reactions they are involved with in columns and elements indicate the level of conservation in each reaction.

A.4.4 Dead reactions

A list of all the reactions that cannot carry flux at steady state (i.e. dead reactions) can be identified from the null space of the stoichiometry matrix or obtained directly from the `model` object:

```
>>> nullspace = stomat.NullSpace()
R1 [1,0]
R2 [1,0]
R3 [1,0]
R4 [1,0]
R5 [1,0]
R6 [1,1]
R7 [2,1]
R8 [0,1]
R9 [0,0]
>>>
>>> dead = model.DeadReactions()
>>> dead
['R9']
```

A.4.5 Enzyme subsets

Enzyme subsets can be obtained by typing:

```
>>> ess = model.EnzSubsets()
```

where `ess` is an instance from the `EnzSubsets`[2] class and can be found in the `ScrumPy/Structural/EnzSubsets` module. A subset can have one of four states assigned to it, they are:

- Dead - reactions in this subset do not carry flux at steady state.
- Irreversible - subset can only carry flux in one direction.
- Reversible - subset can carry flux in either direction.
- Empty - only used during subset calculation and should not appear in final output.

The obtained subsets can be interrogated using Python in-built functions:

```
>>> len(ess)    ##returns total number of subsets
5
>>> str(ess)    ##prints subsets dictionary
"{'R8':{'R8': 1},'Ess_1':{'R4': 1, 'R5': 1,
  'R1': 1,'R2': 1,'R3': 1}, 'R7': {'R7': 1},
  'R6': {'R6': 1}, 'DeadReacs': {'R9': 1}}"
```

[2] subclass of a Python dictionary.

or by attributes attached to the EnzSubsets class:

```
>>> ess.ToList()   ##returns a list of lists
>>>                ##of reactions in each subset
[[‘R1’, ‘R2’, ‘R3’, ‘R4’, ‘R5’], [‘R6’],
[‘R7’], [‘R8’], [(‘R9’, 1)]]
```

A.4.6 Elementary modes

EMs are obtained as an instance of the ModesDB class in the ScrumPy/Structural/ElModes module. The ModesDB class implements simple database functionality for the subsequent interrogation of the set of EMs:

```
>>> elmo = model.ElModes()
>>> len(elmo)             ##number of modes
4
>>> print mo.Modes()
1/2 R4 1/2 R5 1/2 R6 1 R7 1/2 R1 1/2 R2 1/2 R3
  1 R4   1 R5   1 R7 1 R1   1 R2   1 R3   1 R
 -1 R6   1 R4   1 R5    1 R1 1 R2   1 R3   2 R8
 -1 R8   1 R6   1 R7
>>> print mo.Stos()
-1 X_ATP -1/2 X_GLC  1 X_ADP 1 X_DHAP
-2 X_ATP   -1 X_GLC  2 X_ADP 1 X_DHAP 1 X_GAP
-2 X_ATP   -1 X_GLC  2 X_ADP 2 X_GAP
-1 X_GAP    1 X_DHAP
```

where Modes is a ModesDB method that returns a Python string representation of the modes based on reaction names and their respective coefficients. Similarly, Stos prints the EMs in terms of their net external metabolite consumption (negative coefficients) and production (positive coefficients). EMs that do not use any external metabolites (i.e. futile cycles) manifest themselves as empty lines in the Stos output. A separate ModesDB instance of all the futile cycles can be obtained from the original elmo instance:

```
>>> fut = elmo.Futile()
```

For larger models the sheer number of generated modes requires filtering methods to classify modes according to user-defined criteria such as:

```
>>> glu = elmo.Consumes(‘X_GLC’)    ##consume glucose
>>> dha = elmo.Produces(‘X_DHAP’)   ##produce dhap
```

both of which return a ModesDB instance that can be interrogated further. Alternatively, the output from an EMA can also be obtained as an EMs reaction dynamic matrix ($\mathbf{E}_M$) and an EMs stoichiometry dynamic matrix ($\mathbf{E}_S$):

```
>>> em = elmo.mo.Copy()
>>> es = elmo.sto.Copy()
```

em indicates which reactions form a particular EM and their associated fluxes and es shows the net usage of external metabolites.

APPENDIX B

Test Yeast Model in *.spy* Format

Figure B.1 – Anaerobic yeast model in *.spy* format, as described in Section 2.5 and illustrated in Figure 2.8.

```
############################################################
## TEST YEAST ANAEROBIC METABOLISM MODEL                  ##
############################################################

Structural()
External(X1, X2, X3, X4)

############################################################
############################################################

   R1:
       X1 -> S1 ~

   R2:
       S1 <> S2 ~

   R3:
       S2 <> S3 + S4 ~

   R4:
       S3 <> S4 ~

   R5:
       3 S1 -> 2 S2 + S4 + 3 X2 ~

   R6:
       S3 -> X3 ~

   R7:
       S4 -> X4 + X2 ~

############################################################
############################################################
```

APPENDIX C

Minimal *S. erythraea* Model in *.spy* Format

Figure C.1 – Minimal *S. erythraea* model in *.spy* format, as described in Section 3.2 and illustrated in Figure 3.1. See List of Abbreviations section for metabolite abbreviations and Appendix D for a list of enzymes as correlated to the reactions in the description.

```
#############################################################

Structural()
External(GLCx, OXOGx, HCO3, CO2)      ##Objective metabolites
External(H, H2O, Pi)                  ##Exchange metabolites
External(ATP, ADP)                    ##Currency metabolites
External(NADP, NADPH, NAD, NADH, FAD, FADH2)

#############################################################
## GLYCOLYSIS ###############################################
#############################################################

   R1:
       GLCx -> GLC ~

   R2:
       ATP + GLC -> ADP + G6P ~

   R3:
       G6P <> F6P ~

   R4:
       F6P + ATP -> FBP + ADP ~

   R5:
       FBP <> GAP + DHAP ~

   R6:
       DHAP <> GAP ~

   R7:
       GAP + NAD + Pi <> BPG + NADH + H ~

   R8:
       BPG + ADP <> P3G + ATP ~

   R9:
       P3G <> P2G ~

   R10:
       P2G <> PEP + H2O ~

   R11:
       PEP + ADP -> PYR + ATP ~

#############################################################
```

```
#############################################################
## PENTOSE PHOSPHATE PATHWAY ################################
#############################################################

   R12:
      G6P + NADP -> PGL + NADPH + H ~

   R13:
      PGL + H2O <> PGC ~

   R14:
      PGC + NADP -> RU5P + CO2 + NADPH + H ~

   R15:
      RU5P <> R5P ~

   R16:
      RU5P <> X5P ~

   R17:
      X5P + R5P <> GAP + S7P ~

   R18:
      S7P + GAP <> F6P + E4P ~

   R19:
      X5P + E4P <> GAP + F6P ~

#############################################################
## ENTNER-DUORDOFF PATHWAY ##################################
#############################################################

   R20:
      PGC <> KDPG + H2O ~

   R21:
      KDPG <> GAP + PYR ~

#############################################################
## PYRUVATE DEHYDROGENASE COMPLEX ###########################
#############################################################

   R22:
      PYR + NAD + COA <> ACCOA + CO2 + NADH ~

#############################################################
```

```
#############################################################
## TCA CYCLE ################################################
#############################################################

   R23:
      ATP + PYR + HCO3 <> ADP + Pi + OXOA ~

   R24:
      ACCOA + OXOA + H2O <> CIT + COA ~

   R25:
      CIT <> ICIT ~

   R26:
      ICIT + NAD <> OXOG + CO2 + NADH + H ~

   R27:
      OXOG <> OXOGx ~

   R28:
      OXOG + NAD + COA <> SUCCOA + CO2 + NADH + H ~

   R29:
      SUCCOA + Pi + ADP <> SUCC + COA + ATP ~

   R30:
      SUCC + FAD <> FUM + FADH2 ~

   R31:
      FUM + H2O <> MAL ~

   R32:
      MAL + NAD <> OXOA + NADH + H ~

#############################################################
## GLYOXYLATE CYCLE #########################################
#############################################################

   R33:
      ICIT -> SUCC + GLX ~

   R34:
      GLX + ACCOA + H2O -> MAL + COA ~

#############################################################
#############################################################
```

APPENDIX D

Reaction Information for Minimal *S. erythraea* Model

Table D.1 – Reaction subscripts, abbreviations, enzyme names and EC numbers used to reconstruct the minimal *S. erythraea* model, as illustrated in Figure 3.1.

Subscript	Abbreviation	Enzyme Name	EC Number
1	-	glucose transporter	-
2	GK	glucokinase	2.7.1.2
3	GPI	glucose-6-phosphate isomerase	5.3.1.9
4	PFK	6-phosphofructokinase	2.7.1.11
5	FBA	fructose-bisphosphate aldolase	4.1.2.13
6	TPI	triose phospoisomerase	5.3.1.1
7	GAPD	glyceraldehyde 3-phosphate dehydrogenase	1.2.1.12
8	PGK	phosphoglycerate kinase	2.7.2.3
9	PGM	phosphoglycerate mutase	5.4.2.1
10	ENO	enolase	4.2.1.11
11	PYK	pyruvate kinase	2.7.1.40
12	GPDH	glucose-6-phosphate 1-dehydrogenase	1.1.1.49
13	PGLASE	6-phosphogluconolactonase	3.1.1.31
14	PGCDH	6-phosphogluconate dehydrogenase	1.1.1.44
15	R5PISO	ribose 5-phosphate isomerase	5.3.1.6
16	X5PEPI	ribulose-phosphate 3-epimerase	5.1.3.1
17	FTKL	transketolase (S7P reaction)	2.2.1.1
18	TAL	transaldolase	2.2.1.2
19	STKL	transketolase (F6P reaction)	2.2.1.1
20	PGCDT	phosphogluconate dehydratase	4.2.1.12
21	KPGCALD	2-keto-3-deoxy-6-phosphogluconate aldolase	4.1.2.14
22	PDH	pyruvate dehydrogenase complex	1.2.4.1 2.3.1.12
23	PYCAR	pyruvate carboxylase	6.4.1.1
24	CITSYN	citrate synthase	2.3.3.1
25	ACN	aconitase	4.2.1.3
26	ICDH	isocitrate dehydrogenase	1.1.1.42
27	-	2-oxoglutarate transporter	-
28	KDH	2-oxoglutarate dehydrogenase complex	1.2.4.2 2.3.1.61
29	SCOASYN	succinyl-CoA synthetase	6.2.1.5
30	SUCDH	succinate dehydrogenase	1.3.99.1
31	FUMASE	fumarase	4.2.1.2
32	MDH	malate dehydrogenase	1.1.1.37
33	ICLY	isocitrate lyase	4.1.3.1
34	MSYN	malate synthase	2.3.3.9

APPENDIX E

KEGG Organism Abbreviations and Taxonomy Information

Table E.1 – Species names for the KEGG prokaryote abbreviations used in the disseration.

Archaea

mtp	*Methanosaeta thermophila* PT
mbu	*Methanococcoides burtonii* DSM 6242
mba	*Methanosarcina barkeri fusaro*
mac	*Methanosarcina acetivorans* C2A
mma	*Methanosarcina mazei* Go1
mem	*Methanoculleus marisnigri* JR1
mhu	*Methanospirillum hungatei* JF-1
mst	*Methanosphaera stadtmanae* DSM 3091
mth	*Methanothermobacter thermautotrophicus* Delta H
mka	*Methanopyrus kandleri* AV19
mmp	*Methanococcus maripaludis* S2
mmq	*Methanococcus maripaludis* C5
mja	*Methanocaldococcus jannaschii* DSM 2661
afu	*Archaeoglobus fulgidus* DSM 4304
hal	*Halobacterium sp.* NRC-1
hma	*Haloarcula marismortui* ATCC 43049
nph	*Natronomonas pharaonis* DSM 2160
hwa	*Haloquadratum walsbyi* DSM 16790
pto	*Picrophilus torridus* DSM 9790
tac	*Thermoplasma acidophilum* DSM 1728
tvo	*Thermoplasma volcanium* GSS1
tko	*Thermococcus kodakaraensis* KOD1
pfu	*Pyrococcus furiosus* DSM 3638
pho	*Pyrococcus horikoshii* OT3
pab	*Pyrococcus abyssi* GE5
mse	*Metallosphaera sedula* DSM 5348
sso	*Sulfolobus solfataricus* P2
sto	*Sulfolobus tokodaii* 7
sai	*Sulfolobus acidocaldarius* DSM 639
pis	*Pyrobaculum islandicum* DSM 4184
pcl	*Pyrobaculum calidifontis* JCM 11548
pai	*Pyrobaculum aerophilum* IM2
ape	*Aeropyrum pernix* K1
hbu	*Hyperthermus butylicus* DSM 5456

Actinobacteria

cgb	*Corynebacterium glutamicum* ATCC 13032 (Kitasato)
cgl	*Corynebacterium glutamicum* ATCC 13032 (Kitasato)
cef	*Corynebacterium efficiens* YS-314
cdi	*Corynebacterium diphtheriae* NCTC 13129
cjk	*Corynebacterium jeikeium* K411
aau	*Arthrobacter aurescens* TC1
art	*Arthrobacter sp.* FB24

lxx	*Leifsonia xyli xyli* CTCB07
mle	*Mycobacterium leprae* TN
mbb	*Mycobacterium bovis* BCG Pasteur 1173P2
mbo	*Mycobacterium bovis* AF2122/97
mtu	*Mycobacterium tuberculosis* H37Rv
mtc	*Mycobacterium tuberculosis* CDC1551
mav	*Mycobacterium avium* 104
mpa	*Mycobacterium avium paratuberculosis* K-10
mjl	*Mycobacterium sp.* JLS
mkm	*Mycobacterium sp.* KMS
mmc	*Mycobacterium sp.* MCS
mgi	*Mycobacterium gilvum* PYR-GCK
mva	*Mycobacterium vanbaalenii* PYR-1
msm	*Mycobacterium smegmatis* MC2 155
rha	*Rhodococcus sp.* RHA1
nfa	*Nocardia farcinica IFM* 10152
nca	*Nocardioides sp.* JS614
fal	*Frankia alni* ACN14a
fra	*Frankia sp.* CcI3
stp	*Salinispora tropica* CNB-440
sen	*Saccharopolyspora erythraea* NRRL 2338
sco	*Streptomyces coelicolor* A3(2)
sma	*Streptomyces avermitilis* MA-4680
tfu	*Thermobifida fusca* YX
ace	*Acidothermus cellulolyticus* 11B
pac	*Propionibacterium acnes* KPA171202
twh	*Tropheryma whipplei* Twist
tws	*Tropheryma whipplei* TW08/27
blo	*Bifidobacterium longum* NCC2705
rxy	*Rubrobacter xylanophilus* DSM 9941

Firmicutes

tte	*Thermoanaerobacter tengcongensis* MB4
mta	*Moorella thermoacetica* ATCC 39073
csc	*Caldicellulosiruptor saccharolyticus* DSM 8903
swo	*Syntrophomonas wolfei wolfei* Goettingen
chy	*Carboxydothermus hydrogenoformans* Z-2901
drm	*Desulfotomaculum reducens* MI-1
dsy	*Desulfitobacterium hafniense* Y51
cth	*Clostridium thermocellum* ATCC 27405
cpr	*Clostridium perfringens* SM101
cpf	*Clostridium perfringens* ATCC 13124
cpe	*Clostridium perfringens* 13
cno	*Clostridium novyi* NT
ctc	*Clostridium tetani* E88
cac	*Clostridium acetobutylicum* ATCC 824
cdf	*Clostridium difficile* 630
bca	*Bacillus cereus* ATCC 10987

bce	*Bacillus cereus* ATCC 14579
btk	*Bacillus thuringiensis* sv konkukian 97-27
btl	*Bacillus thuringiensis* Al Hakam
bat	*Bacillus anthracis* Sterne
ban	*Bacillus anthracis* Ames
bar	*Bacillus anthracis* Ames Ancestor
bcz	*Bacillus cereus* E33L
bcl	*Bacillus clausii* KSM-K16
bha	*Bacillus halodurans* C-125
bsu	*Bacillus subtilis* 168
bld	*Bacillus licheniformis* ATCC 14580 (Novozymes)
bli	*Bacillus licheniformis* ATCC 14580 (Novozymes)
oih	*Oceanobacillus iheyensis* HTE831
gka	*Geobacillus kaustophilus* HTA426
lwe	*Listeria welshimeri* sv 6b SLCC5334
lin	*Listeria innocua* Clip11262
lmf	*Listeria monocytogenes* 4b F2365
lmo	*Listeria monocytogenes* EGD-e
sav	*Staphylococcus aureus aureus* Mu50
sau	*Staphylococcus aureus aureus* N315
sas	*Staphylococcus aureus aureus* MSSA476
sam	*Staphylococcus aureus aureus* MW2
saa	*Staphylococcus aureus aureus* USA300
sac	*Staphylococcus aureus aureus* COL
sao	*Staphylococcus aureus aureus* NCTC 8325
sar	*Staphylococcus aureus aureus* MRSA252
sab	*Staphylococcus aureus* RF122
sep	*Staphylococcus epidermidis* ATCC 12228
ser	*Staphylococcus epidermidis* RP62A
ssp	*Staphylococcus saprophyticus saprophyticus* ATCC 15305
sha	*Staphylococcus haemolyticus* JCSC1435
efa	*Enterococcus faecalis* V583
lsa	*Lactobacillus sakei sakei* 23K
lpl	*Lactobacillus plantarum* WCFS1
lsl	*Lactobacillus salivarius salivarius* UCC118
ljo	*Lactobacillus johnsonii* NCC 533
lac	*Lactobacillus acidophilus* NCFM
ldb	*Lactobacillus delbrueckii bulgaricus* ATCC 11842
spm	*Streptococcus pyogenes* MGAS8232
sps	*Streptococcus pyogenes* SSI-1
spg	*Streptococcus pyogenes* MGAS315
spy	*Streptococcus pyogenes* M1 GAS
spk	*Streptococcus pyogenes* MGAS9429
spj	*Streptococcus pyogenes* MGAS2096
spi	*Streptococcus pyogenes* MGAS10750
sph	*Streptococcus pyogenes* MGAS10270
spb	*Streptococcus pyogenes* MGAS6180
spz	*Streptococcus pyogenes* MGAS5005
spf	*Streptococcus pyogenes* Manfredo
spa	*Streptococcus pyogenes* MGAS10394

sag	*Streptococcus agalactiae* 2603V/R
san	*Streptococcus agalactiae* NEM316
sak	*Streptococcus agalactiae* A909
spd	*Streptococcus pneumoniae* D39
spr	*Streptococcus pneumoniae* R6
spn	*Streptococcus pneumoniae* TIGR4 454
ssa	*Streptococcus sanguinis* SK36
stl	*Streptococcus thermophilus* LMG 18311
stc	*Streptococcus thermophilus* CNRZ1066
smu	*Streptococcus mutans* UA159
llm	*Lactococcus lactis cremoris* MG1363
sth	*Symbiobacterium thermophilum* IAM 14863
mcp	*Mycoplasma capricolum capricolum* ATCC 27343
mmy	*Mycoplasma mycoides mycoides* SC PG1
mhy	*Mycoplasma hyopneumoniae* 232
mhj	*Mycoplasma hyopneumoniae* J
mhp	*Mycoplasma hyopneumoniae* 7448
mpu	*Mycoplasma pulmonis* UAB CTIP
msy	*Mycoplasma synoviae* 53
mmo	*Mycoplasma mobile* 163K
mpn	*Mycoplasma pneumoniae* M129
mge	*Mycoplasma genitalium* G37 454
mga	*Mycoplasma gallisepticum* R
mpe	*Mycoplasma penetrans* HF-2
uur	*Ureaplasma parvum sv* 3 ATCC 700970
mfl	*Mesoplasma florum* L1
ayw	*Aster yellows witches-broom phytoplasma* AYWB

α-proteobacteria

rco	*Rickettsia conorii* Malish 7
rfe	*Rickettsia felis* URRWXCal2
rbe	*Rickettsia bellii* RML369-C
rty	*Rickettsia typhi* Wilmington
rpr	*Rickettsia prowazekii* Madrid E
wol	*Wolbachia* endosymbiont of Drosophila melanogaster
wbm	*Wolbachia* endosymbiont TRS of Brugia malayi
nse	*Neorickettsia sennetsu* Miyayama
erg	*Ehrlichia ruminantium* Gardel
eru	*Ehrlichia ruminantium* Welgevonden (CIRAD)
erw	*Ehrlichia ruminantium* Welgevonden (CIRAD)
ecn	*Ehrlichia canis* Jake
ech	*Ehrlichia chaffeensis* Arkansas
aph	*Anaplasma phagocytophilum* HZ
ama	*Anaplasma marginale* St. Maries
pub	*Candidatus Pelagibacter* ubique HTCC1062
bqu	*Bartonella quintana* Toulouse
bhe	*Bartonella henselae* Houston-1
bbk	*Bartonella bacilliformis* KC583

nwi	*Nitrobacter winogradskyi* Nb-255
nha	*Nitrobacter hamburgensis* X14
rpc	*Rhodopseudomonas palustris* BisB18
rpe	*Rhodopseudomonas palustris* BisA53
rpd	*Rhodopseudomonas palustris* BisB5
rpb	*Rhodopseudomonas palustris* HaA2
rpa	*Rhodopseudomonas palustris* CGA009
bja	*Bradyrhizobium japonicum* USDA 110
atc	*Agrobacterium tumefaciens* C58 (Dupont)
atu	*Agrobacterium tumefaciens* C58 (Dupont)
sme	*Sinorhizobium meliloti* 1021
rle	*Rhizobium leguminosarum bv.* viciae 3841
ret	*Rhizobium etli* CFN 42
mlo	*Mesorhizobium loti* MAFF303099
mes	*Mesorhizobium sp.* BNC1
bms	*Brucella suis* 1330
bmb	*Brucella abortus* bv 1 9-941
bmf	*Brucella melitensis* bv Abortus 2308
bme	*Brucella melitensis* 16M
mag	*Magnetospirillum magneticum* AMB-1
rru	*Rhodospirillum rubrum* ATCC 11170
gox	*Gluconobacter oxydans* 621H
gbe	*Granulobacter bethesdensis* CGDNIH1
pde	*Paracoccus denitrificans* PD1222
rsq	*Rhodobacter sphaeroides* ATCC 17025
rsh	*Rhodobacter sphaeroides* ATCC 17029
rsp	*Rhodobacter sphaeroides* 2.4.1
jan	*Jannaschia sp.* CCS1
rde	*Roseobacter denitrificans* OCh 114
sil	*Silicibacter pomeroyi* DSS-3
sit	*Silicibacter sp.* TM1040
hne	*Hyphomonas neptunium* ATCC 15444
mmr	*Maricaulis maris* MCS10
ccr	*Caulobacter crescentus* CB15
nar	*Novosphingobium aromaticivorans* DSM 12444
sal	*Sphingopyxis alaskensis* RB2256
eli	*Erythrobacter litoralis* HTCC2594
zmo	*Zymomonas mobilis mobilis* ZM4

β-proteobacteria

bpa	*Bordetella parapertussis* 12822
bbr	*Bordetella bronchiseptica* RB50
bpe	*Bordetella pertussis* Tohama I
bxe	*Burkholderia xenovorans* LB400
bam	*Burkholderia cepacia* AMMD
bch	*Burkholderia cenocepacia* HI2424
bcn	*Burkholderia cenocepacia* AU 1054
bur	*Burkholderia sp.* 383

bvi	*Burkholderia vietnamiensis* G4
bps	*Burkholderia pseudomallei* K96243
bpm	*Burkholderia pseudomallei* 1710b
bpd	*Burkholderia pseudomallei* 668
bmv	*Burkholderia mallei* SAVP1
bma	*Burkholderia mallei* ATCC 23344
bml	*Burkholderia mallei* NCTC 10229
bmn	*Burkholderia mallei* NCTC 10247
bte	*Burkholderia thailandensis* E264
rso	*Ralstonia solanacearum* GMI1000
reh	*Ralstonia eutropha* H16
rme	*Ralstonia metallidurans* CH34
reu	*Ralstonia eutropha* JMP134
pnu	*Polynucleobacter sp.* QLW-P1DMWA-1
har	*Herminiimonas arsenicoxydans*
vei	*Verminephrobacter eiseniae* EF01-2
rfr	*Rhodoferax ferrireducens* T118
pna	*Polaromonas naphthalenivorans* CJ2
pol	*Polaromonas sp.* JS666
ajs	*Acidovorax sp.* JS42
mpt	*Methylibium petroleiphilum* PM1
azo	*Azoarcus sp.* BH72
eba	*Azoarcus sp.* EbN1
dar	*Dechloromonas aromatica* RCB
cvi	*Chromobacterium violaceum* ATCC 12472
nma	*Neisseria meningitidis* Z2491
nme	*Neisseria meningitidis* MC58
nmc	*Neisseria meningitidis* FAM18
ngo	*Neisseria gonorrhoeae* FA 1090
mfa	*Methylobacillus flagellatus* KT
tbd	*Thiobacillus denitrificans* ATCC 25259
net	*Nitrosomonas eutropha* C71
neu	*Nitrosomonas europaea* ATCC 19718
nmu	*Nitrosospira multiformis* ATCC 25196

γ-proteobacteria

ecs	*Escherichia coli* O157:H7 Sakai
ece	*Escherichia coli* O157:H7 EDL933
ecj	*Escherichia coli* W3110
eco	*Escherichia coli* K12
ssn	*Shigella sonnei* Ss046
ecp	*Escherichia coli* 536
ecc	*Escherichia coli* CFT073
ecv	*Escherichia coli* APEC O1
eci	*Escherichia coli* UTI89
sbo	*Shigella boydii* Sb227
sfl	*Shigella flexneri* 2a 301
sfx	*Shigella flexneri* 2a 2457T

sdy	*Shigella dysenteriae* Sd197
sec	*Salmonella enterica enterica sv* Choleraesuis SC-B67
stm	*Salmonella typhimurium* LT2
spt	*Salmonella enterica enterica sv* Paratyphi A ATCC 9150
sty	*Salmonella enterica enterica sv* Typhi CT18
stt	*Salmonella enterica enterica sv* Typhi Ty2
ent	*Enterobacter sp.* 638
ypp	*Yersinia pestis* Pestoides F
yps	*Yersinia pseudotuberculosis* IP 32953
ypn	*Yersinia pestis* Nepal516
ypm	*Yersinia pestis* biovar Microtus 91001
ypk	*Yersinia pestis* KIM
ype	*Yersinia pestis* CO92
ypa	*Yersinia pestis* Antiqua
eca	*Erwinia carotovora* atroseptica SCRI1043
plu	*Photorhabdus luminescens* laumondii TTO1
sgl	*Sodalis glossinidius* morsitans
wbr	*Wigglesworthia glossinidia* endosymbiont of Glossina brevipalpis
bpn	*Candidatus Blochmannia pennsylvanicus* BPEN
bfl	*Candidatus Blochmannia floridanus*
bcc	*Buchnera aphidicola* Cc
bab	*Buchnera aphidicola* Bp
buc	*Buchnera aphidicola* APS
bas	*Buchnera aphidicola* Sg
vvu	*Vibrio vulnificus* CMCP6
vvy	*Vibrio vulnificus* YJ016
vpa	*Vibrio parahaemolyticus* RIMD 2210633
vch	*Vibrio cholerae* O1 bv eltor N16961
vco	*Vibrio cholerae* O395
vfi	*Vibrio fischeri* ES114
ppr	*Photobacterium profundum* SS9
aha	*Aeromonas hydrophila hydrophila* ATCC 7966
pin	*Psychromonas ingrahamii* 37
sdn	*Shewanella denitrificans* OS217
sfr	*Shewanella frigidimarina* NCIMB 400
saz	*Shewanella amazonensis* SB2B
slo	*Shewanella loihica* PV-4
sbl	*Shewanella baltica* OS155
spc	*Shewanella putrefaciens* CN-32
shw	*Shewanella sp.* W3-18-1
son	*Shewanella oneidensis* MR-1
shn	*Shewanella sp.* ANA-3
shm	*Shewanella sp.* MR-7
she	*Shewanella sp.* MR-4
sde	*Saccharophagus degradans* 2-40
maq	*Marinobacter aquaeolei* VT8
pha	*Pseudoalteromonas haloplanktis* TAC125
pat	*Pseudoalteromonas atlantica* T6c
ilo	*Idiomarina loihiensis* L2TR
cps	*Colwellia psychrerythraea* 34H

csa	*Chromohalobacter salexigens* DSM 3043
hch	*Hahella chejuensis* KCTC 2396
abo	*Alcanivorax borkumensis* SK2
aci	*Acinetobacter sp.* ADP1
pcr	*Psychrobacter cryohalolentis* K5
psp	*Pseudomonas syringae pv.* phaseolicola 1448A
psb	*Pseudomonas syringae pv.* syringae B728a
pst	*Pseudomonas syringae pv.* tomato DC3000
pmy	*Pseudomonas mendocina* ymp
ppu	*Pseudomonas putida* KT2440
pen	*Pseudomonas entomophila* L48
pae	*Pseudomonas aeruginosa* PAO1
pfo	*Pseudomonas fluorescens* PfO-1
pfl	*Pseudomonas fluorescens* Pf-5
hha	*Halorhodospira halophila* SL1
aeh	*Alkalilimnicola ehrlichei* MLHE-1
noc	*Nitrosococcus oceani* ATCC 19707
mca	*Methylococcus capsulatus* Bath
xcv	*Xanthomonas campestris pv.* vesicatoria 85-10
xac	*Xanthomonas axonopodis pv.* citri 306
xcc	*Xanthomonas campestris pv.* campestris ATCC 33913
xcb	*Xanthomonas campestris pv.* campestris 8004
xoo	*Xanthomonas oryzae pv.* oryzae KACC10331
xft	*Xylella fastidiosa* Temecula1
xfa	*Xylella fastidiosa* 9a5c
hdu	*Haemophilus ducreyi* 35000HP
hso	*Haemophilus somnus* 129PT
hin	*Haemophilus influenzae* Rd KW20
hit	*Haemophilus influenzae* 86-028NP
msu	*Mannheimia succiniciproducens* MBEL55E
apl	*Actinobacillus pleuropneumoniae* L20
pmu	*Pasteurella multocida multocida* Pm70
lpn	*Legionella pneumophila pneumophila* Philadelphia 1
lpp	*Legionella pneumophila* Paris
lpf	*Legionella pneumophila* Lens
cbu	*Coxiella burnetii* RSA 493
tcx	*Thiomicrospira crunogena* XCL-2
ftl	*Francisella tularensis holarctica*
fth	*Francisella tularensis holarctica* OSU18
ftw	*Francisella tularensis tularensis* WY96-3418
ftf	*Francisella tularensis tularensis* FSC 198
ftu	*Francisella tularensis tularensis* SCHU S4
rma	*Candidatus Ruthia* magnifica Cm
bci	*Baumannia cicadellinicola* Hc

APPENDIX F

Publication