Technical Report

UCAM-CL-TR-945
ISSN 1476-2986

Number 945

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Machine learning methods for detecting structure in metabolic flow networks

Maxwell Jay Conway

March 2020

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted August 2018 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Selwyn College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Abstract

Machine learning methods for detecting structure in metabolic flow networks

Maxwell Jay Conway

Metabolic flow networks are large scale, mechanistic biological models with good predictive power. However, even when they provide good predictions, interpreting the meaning of their structure can be very difficult, especially for large networks which model entire organisms. This is an underaddressed problem in general, and the analytic techniques that exist currently are difficult to combine with experimental data. The central hypothesis of this thesis is that statistical analysis of large datasets of simulated metabolic fluxes is an effective way to gain insight into the structure of metabolic networks. These datasets can be either simulated or experimental, allowing insight on real world data while retaining the large sample sizes only easily possible via simulation. This work demonstrates that this approach can yield results in detecting structure in both a population of solutions and in the network itself.

This work begins with a taxonomy of sampling methods over metabolic networks, before introducing three case studies, of different sampling strategies. Two of these case studies represent, to my knowledge, the largest datasets of their kind, at around half a million points each. This required the creation of custom software to achieve this in a reasonable time frame, and is necessary due to the high dimensionality of the sample space.

Next, a number of techniques are described which operate on smaller datasets. These techniques, focused on pairwise comparison, show what can be achieved with these smaller datasets, and how in these cases, visualisation techniques are applicable which do not have simple analogues with larger datasets.

In the next chapter, Similarity Network Fusion is used for the first time to cluster organisms across several levels of biological organisation, resulting in the detection of discrete, quantised biological states in the underlying datasets. This quantisation effect was maintained across both real biological data and Monte-Carlo simulated data, with

related underlying biological correlates, implying that this behaviour stems from the network structure itself, rather than from the genetic or regulatory mechanisms that would normally be assumed.

Finally, Hierarchical Block Matrices are used as a model of multi-level network structure, by clustering reactions using a variety of distance metrics: first standard network distance measures, then by Local Network Learning, a novel approach of measuring connection strength via the gain in predictive power of each node on its neighbourhood. The clusters uncovered using this approach are validated against pre-existing subsystem labels and found to outperform alternative techniques.

Overall this thesis represents a significant new approach to metabolic network structure detection, as both a theoretical framework and as technological tools, which can readily be expanded to cover other classes of multilayer network, an under explored datatype across a wide variety of contexts. In addition to the new techniques for metabolic network structure detection introduced, this research has proved fruitful both in its use in applied biological research and in terms of the software developed, which is experiencing substantial usage.

Acknowledgements

First of all, thanks to Pietro Lió, for his exemplary support, guidance and encouragement. I would also like to thank the rest of the group, particularly Claudio Angione, for numerous ideas and excellent advice.

Thanks to my various collaborators throughout the PhD, not only for the data they provided but also for the huge amount I learnt from them, and thanks to the EPSRC—without their financial support this would not have been possible.

And finally, my deepest thanks to my family and friends, both new and old, for their invaluable help and support, both intellectual and emotional.

Contents

CHAPTER 1

Introduction

1.1 Overview

Metabolic networks are one of the most successful approaches to modelling how living cells work. They are capable of making verifiably correct predictions [1] from mechanistic simulations of whole cell biochemistry, and are used extensively in academic and industrial biotechnology [2]. However, with hundreds [3–5] of these models available, each with hundreds or thousands of reactions, approaches to generating real understanding are lagging behind black box predictive power.

This work introduces and demonstrates a number of statistical and machine learning approaches to network structure detection. The central hypothesis is that inference from large datasets of feasible metabolic flow configurations is an effective way to derive information about the structure of a metabolic network. Specifically, the simulated datasets used here are of optimal networks, which have been optimised by linear programming under some set of conditions, based on small perturbations around a high biomass value. This is slower than uniform sampling, but I argue in Chapter 3 that it is more biologically justifiable, and it is more directly comparable with techniques used on experimental data. As well as being relatively under explored in the context of metabolic flow networks, these techniques have the distinct advantage that they can be applied to both real data and Monte Carlo simulations even where the properties of the input data are poorly understood or characterised.

Chapter 2, *Background*, describes biological background, such as the shape, properties, and reasonable assumptions about metabolic networks, and why these properties make them amenable to computational simulation and analysis. It also describes technological background: network representations, relevant mathematical theory, and related works.

Chapter 3, *Characterising state spaces of flow networks: sampling metabolic models*, describes a taxonomy of methods for Monte Carlo sampling on metabolic networks, or-

ganised by frameworks for methods of modifying models and sources for distributions to bias the modifications selected. It then finishes with a selection of example case studies of sampling strategies, which are used as demonstration data in Chapter 5 and Chapter 6.

Chapter 4, *Small sample approaches*, describes three different approaches to creation and interpretation of small sample sizes of metabolic networks in concrete biological settings. A central thread of this chapter is that all three methods attempt different ways of understanding metabolic networks visually, and while these have their strengths, they also demonstrate the need for statistical techniques that can detect network structures. This forms something of a counterpoint to the methods in chapters 5 and 6, which deal with much larger sample sizes.

Chapter 5, *Network interpretation by Similarity Network Fusion*, describes the application of the Similarity Network Fusion [6] technique to metabolic datasets. This is applied to both a real world gene expression dataset and to simulated data, in order to show patterns in the samples in both datasets. The results are clustered, resulting in the types of phenotype groupings which would typically be attributed to genetic variation. However, the results are similar across both experimental and simulated datasets, and have related underlying causes, implying that these clusters in fact stem from network structure itself.

Chapter 6, *Hierarchical Block Matrices and Local Network Learning* describes a project to detect and uncover latent structure in flow networks by combining biased Monte Carlo sampling with a local network approach to unsupervised detection of important network links. This is applied to detect network structure in both real and simulated datasets, and is validated by comparison with pre-existing manual subsystem labels. The results are shown to be significantly better predictors of subsystem labelling than would be expected by chance, and to outperform off the shelf clustering techniques.

Figure 1.1 shows some of the links between chapters in terms of data or techniques. This emphasises the overall structure of the thesis: Chapter 3 focusses on techniques and examples of data generation, while chapters 4, 5 and 6 introduce techniques and examples of data interpretation.

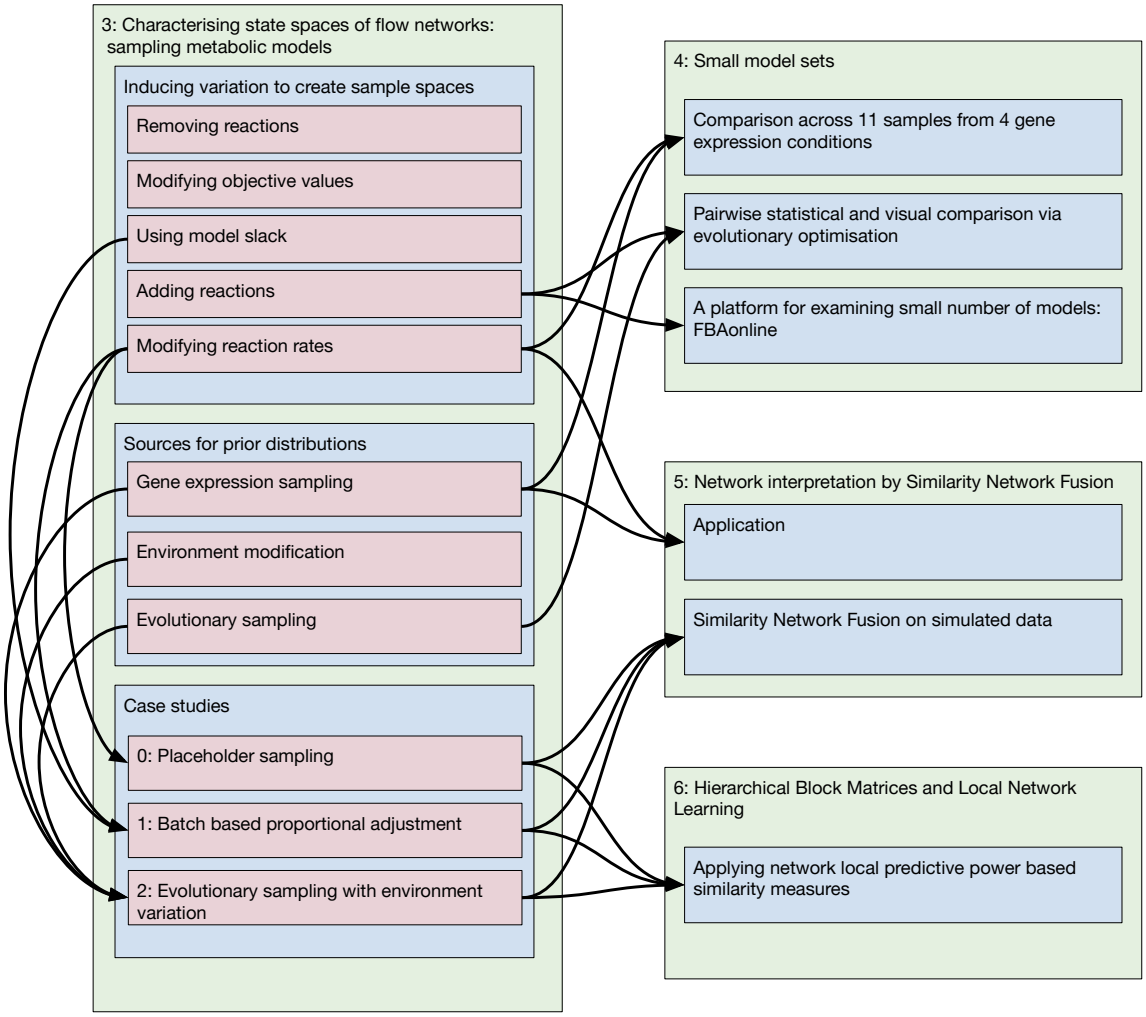


Figure 1.1: Diagram of links between chapters. Arrows represent links of either data usage or technique usage: where the data and techniques introduced in Chapter 3 are used in the other chapters. Green boxes represent chapters, blue boxes represent sections, red boxes represent subsections and so on. Sections on theory or implementation which are not shared between chapters are omitted for brevity.

1.2 Papers

The following is a list of publications that I have been involved with throughout my PhD. Of these, [7, 8] are particularly relevant to Chapter 2, [9–11] are particularly relevant to Chapter 4, and Chapter 5 is based on work in [12]. All work covered in this thesis is wholly my own, along with the associated portions of these papers, though other parts of the papers were done in collaboration.

- [13]: Bioinformatics challenges and potentialities in studying extreme environments. Claudio Angione, Pietro Li, Sandra Pucciarelli, Basarbatu Can, Maxwell Conway, Marina Lotti, Habib Bokhari, Alessio Mancini, Ugur Sezerman, Andrea Telatin. International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics, pages 205–219, 2015. In this paper I advised on the metabolic networking content.

- [12]: Multiplex methods provide effective integration of multi-omic data in genome-scale models. Claudio Angione, Maxwell Conway, Pietro Lió. BMC Bioinformatics, 2016. In this paper I contributed all of the work on Similarity Network Fusion, while the simulation component was done in collaboration. Since then I have reimplemented the simulation component myself for convenience, as is described in Chapter 5.

- [10]: Iterative Multi Level Calibration of Metabolic Networks. Maxwell Conway, Claudio Angione, Pietro Lió. Current Bioinformatics, 2016. In this I created the multi-objective optimization procedure (though starting from an existing code base described in [14]), and did all other work: creating the visualisation framework described and all scratch.

- [7]: Seeing the wood for the trees: a forest of methods for optimization and omic-network integration in metabolic modelling. Supreeta Vijayakumar, Maxwell Conway, Pietro Lió, Claudio Angione. Briefings in bioinformatics, 2017. To this review I contributed the tutorial section, and some of the body text, and two of the three figures.

- [9]: Transcriptome and proteome analysis of *Salmonella enterica* serovar Typhimurium systemic infection of wild type and immune-deficient mice. Olusegun Oshota, Maxwell Conway, Maria Fookes, Fernanda Schreiber, Roy R Chaudhuri, Lu Yu, Fiona JE Morgan, Simon Clare, Jyoti Choudhuri, Nicholas R Thomson and others. PloS one, 2017. In this paper I contributed all work involving metabolic modelling: 5 of the written sections and figures 4 to 6.

- [8]: Optimization of Multi-Omic Genome-Scale Models: Methodologies, Hands-on Tutorial, and Perspectives. Supreeta Vijayakumar, Maxwell Conway, Pietro Lió, Claudio Angione. Metabolic Network Reconstruction and Modeling, pages 389–408, 2018. I contributed the perspective in section 4 to this background piece.

- [11]: CiliateGEM: an open-project and a tool for predictions of ciliate metabolic variations and experimental condition design. Alessio Mancini, Filmon Eyassu, Maxwell Conway, Annalisa Occhipinti, Pietro Liò, Claudio Angione, Sandra Pucciarelli. BMC bioinformatics, 2018. In this project I created tools that were used for sections 3 and 5 of the CiliateGEM pipeline in figure 1.

- [15]: STAble: a novel approach to de novo assembly of RNA-seq data and its application in a metabolic model network based metatranscriptomic workflow. Igor Saggese, Elisa Bona, Maxwell Conway, Francesco Favero, Marco Ladetto, Pietro Liò, Giovanni Manzini and Flavio Mignone. BMC bioinformatics, 2018. For this paper I provided advice and discussion.

CHAPTER 2

Background

2.1 Introduction

This chapter describes biological background, such as the shape, properties, and reasonable assumptions about metabolic networks, and why they are amenable to computational simulation and analysis. It also describes technological background: network representations, relevant mathematical theory and related work.

This chapter begins with biological background covering the motivation and justification for the techniques used here (sections 2.2, 2.3 and 2.4), before moving on to an overview of comparable techniques in section 2.5. Technological background is discussed in sections 2.6 and 2.7, before concluding with a detailed review of related techniques in 2.8. Background which is exclusively relevant to particular analytic techniques is left to the appropriate chapters. Particular emphasis is placed on areas that are relevant to simulation based, rather than analytic, techniques for network structure detection, since this is the focus of this thesis.

2.2 A simplified view on how cells work

Humans, like most animals, solve many of our problems by mechanics. If we do not like something, we move away from it. If we are hungry, we go to a place that has food, find it, and eat it. However, this gives us something of a skewed view on what life is: for instance, children often do not immediately think of plants as alive.

Life is primarily about chemistry. Living organisms are cells (occasionally more or less cooperating colonies of more than one cell, like us), and a cell is a bag of chemicals. These chemicals pull outside chemicals into the bag, convert them into useful materials, and expel unwanted by-products. They then use the acquired building materials to construct more biomass (the materials that make them up) until they get large enough to split, and

continue growing.

Most of the molecules in a cell are broadly described as metabolites. These are all the small molecules which form the feedstocks, intermediaries and by-products, for growth within the cell. A good way of thinking of this is that this includes almost everything that one would learn about in high school chemistry, rather than in biology. For instance, water, oxygen, hydrogen ions, salts, sugars, and alcohols.

The heavy lifting of these chemical processes is done by proteins. These are built from long chains of amino acids, connected end to end, which fold and coil into useful shapes that allow them to perform useful functions. Of particular interest are enzymes, which are proteins which facilitate chemical reactions between metabolites, and transporter proteins, which form tunnels and pumps to move metabolites into and out of the cell. Proteins often perform these functions in surprisingly mechanical manners: enzymes are typically shaped to bind loosely to their substrates in such a way as they are forced together to react before being released, whilst active transporters typically operate via a ratcheting principal.

Proteins are large molecules composed from a small set of unique amino acids (20 in humans), chained together. The information on what sequences of amino acids make up each protein is stored in DNA. Once again, this is a chain of subunits from a small alphabet, though in this case there are only four symbols, known as nucleotides, which are interpreted with groups of three encoding each protein.

The process of constructing proteins from the DNA template is known as 'gene expression'. (Potentially confusingly, 'gene expression' also refers to the quantitative rate of this process.) It works as follows:

1. A protein complex called RNA polymerase attaches itself to the DNA strand, splitting open the double helix to read one side.

2. The RNA polymerase protein moves along the strand. At each DNA nucleotide, it attaches a complementary nucleotide of mRNA (mRNA is much like DNA, but more reactive and less stable). It then binds the mRNA nucleotide to the previous mRNA nucleotide.

3. When the RNA polymerase finishes its work, the result is an mRNA negative of the gene that is required.

4. A large protein called a ribosome attaches to the mRNA negative to read it into a protein.

5. Once again, the ribosome moves along the mRNA, this time attaching matching tRNA bases. The tRNA is present in groups of three bases, with each group attached

to one amino acid. The ribosome moves along the mRNA, using the tRNA to attach the correct chain of amino acids together, to create the required protein.

6. The amino acids in the chain curl up to bind to each other to form the correct three dimensional shape.

This process appears quite simple when described in such general terms, however it involves a very large amount of detail and incidental complexity, as described in section 2.3.

2.3 How to model living cells

The mechanisms of gene expression described in the previous section are conceptually fairly simple. In many ways, the manufacturing of proteins is simpler than the manufacturing of many consumer goods—although proteins are far more complex and intricate end results, the process of instantiating them from their 'blueprint' has quite uniform steps.

However, if we want to model how living things behave, we need to know how their metabolism acts, since this is the primary way that most of them interact with the outside world. We also need to know this in a quantitative manner, and this is where we run into serious complexity.

Because the individual mechanisms are generally well understood and relatively simple, it is possible to construct differential equations that accurately describe each part of the process in isolation. However, once these equations are connected together the system explodes in complexity due to the many, many feedback loops. These feedback loops make characterising the system very difficult because it is difficult to experiment on parts in isolation, and because attempts to isolate subsystems *in vitro* inevitably alter the environment. Furthermore, for *in vivo* experiments, measurement and perturbation are difficult. Cells are small, so normally measurements can only be taken on large numbers simultaneously, meaning that only an aggregate result can be found. Because the result is aggregated across many different cells, any effect must be large in order to be detectable, and so the perturbation must be large to generate a detectable result, and to overcome the cell's attempts to counteract it.

Furthermore, because life is evolved rather than designed, much of the regulation and control occurs via what could be described as inelegant methods, which represent local fitness maxima. As an example, if there is too much of a metabolite X, the amount could be reduced by any combination of a vast number of more or less direct methods. For instance, the amount of the enzyme that creates X could be reduced by interfering at any point in its transcription process, or by blocking it from acting, or the opposite could be done to an enzyme that breaks it down, or alternatively any of these actions could

be conducted on an indirectly connected enzyme which acts on a different metabolite to create a knock on effect on the production or degradation of X, or its uptake or removal. Or, indeed, nothing could be done at all, and the cell could perform some other action to just cope with the excess of X. In a designed system, one would hope that there would be a deliberate choice of the most simple and direct method to achieve an end. However in an evolved system the pattern seems to be that the first mutation that helps with a problem is selected for, and any resulting side effects are dealt with later.

2.3.1 Approaches to modelling cells: which black box to choose

When faced with a complicated system, it is common to start by choosing some components to model mechanistically, and some to model as a black box, finding the easiest model that fits the data under the relevant circumstances.

Taking this approach, systems biology has tended towards an overall view that considers the cell as a set of conceptual layers, termed 'omes'. This naming is from the genome, the set of all genes in the cell.

Unlike the other 'omes, the genome is generally static, and when changes do happen these are normally random mutations during copying, rather than any kind of directed change, or short term feedback. This means that most or all of the information about how an organism works must be included in the genome. Fortunately, the stability, structural uniformity and random mutations of the genome also make it easy to study. If you observe a property of an organism and later sequence its genome, then you know that the genome you find was the same as the genome at the time of observation, and that it is extremely unlikely that a genetic feature is a result of some other property of the organism. In the past, particularly prior to the human genome project, there was a hope that if we sequenced enough genomes and compared them with enough phenotypes (the external properties of an organism), then we would be able to infer the genotype-phenotype relationship black box, and that this would give us a predictive model of how life worked.

It should come as no surprise to those from a computer science background that this did not work. The genotype-phenotype relationship is stochastic and stateful. It has as an input tens of thousands of genes (ignoring the vast number of variations that they can take), and has as an output another incredibly complex relationship: a mapping from a vast number of poorly defined aspects of the environment to an equally vast set of potential phenotypic effects. Attempting to understand this kind of relationship without looking at the underlying mechanisms is therefore intractable in the general case.

2.3.2 Approaches to modelling cells: the metabolome is relatively tractable

In the previous section, the problems associated with trying to model the genotype phenotype relationship as a black box were discussed. Of course, the alternative to this approach is to take a more reductionist and mechanistic approach, starting our understanding with simple, well understood subsystems and slowly expanding our understanding outwards.

The metabolome has a number of properties that make it a good area to begin:

- Small molecules are generally completely identical to each other. This contrasts with larger structures like proteins where small variations cannot be individually tracked, since this would lead to a combinatorial explosion.
- The molecules in the metabolome are generally present in large enough quantities that quantisation effects can be ignored.
- Because the molecules of the metabolism are small, they diffuse quickly and their orientation can be ignored.
- The interactions between many groups of small molecules have already been well studied by organic chemists *in vitro*, so that there is already theory with good predictive power, at least until enzymes get involved.
- As discussed previously, chemical exchange is one of the main ways that small organisms interact with the outside world, so metabolic models accurately capture much of their phenotype.

The gold standard in describing metabolic subsystems are systems of partial differential equations. Unfortunately, it is complex and difficult to accurately measure the reaction rate coefficients required, and these models explode in complexity as more and more elements are added and subsystems are joined. This means that if we want to model the complete metabolism of even a very simple organism, we need to use models with less parameters.

2.4 Metabolic modelling

Rate coefficients to chemical reactions can be difficult to measure, especially when they are enzyme mediated, since many, many factors can influence the abundance and activity of enzymes. However, the stoichiometry (the participants and their ratios) of chemical reactions is fixed, since this is determined by conservation of matter and of charge.

Lists have been constructed that detail the stoichiometry of biological reactions, with coverage of most of the reactions that occur in most organisms. Reactions that are

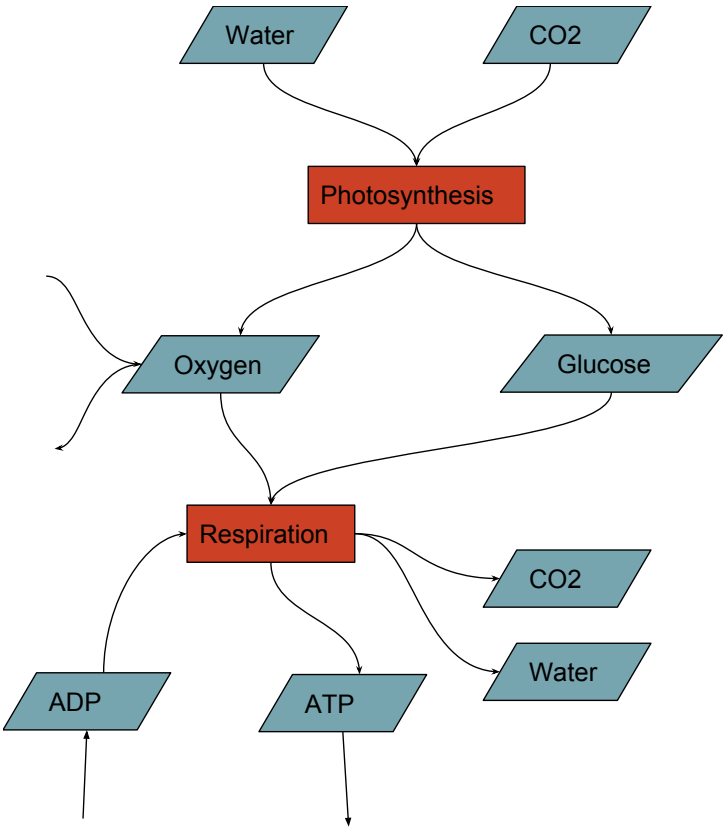


Figure 2.1: A simplified sketch of a metabolic network. In this sketch, we can see that if glucose is conserved and in steady state, the rates of photosynthesis and respiration must be equal. Glucose, water, and CO_2 are examples of metabolites, whilst photosynthesis and respiration are examples of reactions.

missing tend to be unique to particular groups and only used under specific circumstances, such as breaking down unusual foodstuffs or providing protection from toxins, so a large proportion of the reactions occurring in common organisms in benign environments can be covered.

These sets of reactions can be considered as flow networks. As discussed in section 2.7.3, there are a few different ways in which these can be represented, but the simplest model (aside from those that discard information) is to consider the relationship between reactions and metabolites as a bipartite graph: both reactions and metabolites are nodes, with reaction nodes only having edges to metabolite nodes, and vice versa. Edges are directed to represent either consumption or production of a metabolite, and numbered with the stoichiometry of the link. Reactions have a rate, and the rate at which they produce a particular product is their own intrinsic rate, multiplied by the stoichiometry of the link to the product.

From this position, we can start adding assumptions. The first of these is a steady state assumption. This says that there is no net surplus or deficit of any metabolite, so the total rate of production of a metabolite is equal to the total rate of consumption.

This is equivalent to Kirchoff's current law in electronics [16], and means that the rates of the reactions producing the metabolite are tied to the rates of the reactions consuming it. Obviously to allow uptake and excretion this assumption must be relaxed in some areas, which is normally achieved by one sided placeholder reactions. We also add some constraints on rates of reactions. Typically these are quite liberal except for the uptake and excretion placeholders, which we generally assume to be the among the limiting points in the system.

At this stage, we have what is known as a flux space. This is a space with a number of dimensions equal to the number of reactions, and a polytope within this space enclosing the feasible region—the set of reaction rate assignments that are consistent with the assumptions, consistent with the constraints, and consistent with all of the other reaction rates. Figure 2.1 shows a simple example of a metabolic network, showing the different node types and resultant constraints.

If we stop here and directly explore the still under-constrained flux space, we have what are known as unbiased methods. These are decomposition methods that project the full set of dimensions of the flux space down onto a somewhat smaller set of dimensions that cover only the feasible flux space. Unbiased analytic methods such as elementary flux modes are the state of the art in analytic understanding of flow networks, but they have a number of disadvantages in the analysis of large metabolic networks:

1. in typical applications, whilst the dimension reductions are significant, they are still not enough to enable human interpretation,
2. the assumption of a uniform and unbiased flux space is not necessarily reasonable,
3. these approaches are difficult to apply to multi-level models or to real data, and
4. despite being analytic rather than simulation based, they are highly computationally demanding.

Of course, there are various approaches to address each of the weaknesses of unbiased analytic methods, but this thesis eschews these methods in favour of statistical and machine learning post-processing of flux data sets, since this affords increased flexibility and the ability to apply techniques uniformly to a variety of types of models, simulations, and real world data.

If we attempt to find individual solutions within the flux space that are particularly biologically likely, or important, then we have biased methods.

Biased and unbiased methods are both described in more detail in section 2.5 on the next page.

There are many biased methods, but the prototypical method, on which most others are based, is Flux Balance Analysis (FBA). This adds one extra assumption to the model

previously outlined: that evolution has already optimised the organism to grow as fast as possible. To use this assumption we add another placeholder reaction to the existing uptake and excretion nodes—a sink for biomass. This reaction simulates the sequestration of materials that is required for growth, and is generally a reaction with a very large number of input reactants. Unusually the biomass reaction typically has fractional stoichiometries, since we can find the ratios of materials required for growth by simply analysing the constituents of a whole cell.

Once we have a biomass equation, we have a direction for optimisation. We find the assignment of fluxes that will achieve the highest biomass production. This is normally achieved by Linear Programming, which is a fast, specialised optimisation method that is applicable to this type of problem. In small models, this optimisation process can find a single flux assignment, finding a single value for all reactions, but in larger models this is not always the case, as discussed in section 3.2.6 on page 45.

2.5 Overview of existing metabolic network analysis techniques

The most common and in many ways best model for a system of chemical reactions is a system of differential equations. The problem is that building such a system requires knowledge of rate coefficients for every reaction, which are difficult to measure, especially for enzyme catalysed reactions *in vivo*. This makes that approach intractable for full organism models.

Therefore, techniques to analyse large metabolic models must get by with just stoichiometry and various other constraints on reaction rates, such as constant bounds. These methods can be divided into two groups [17]: biased methods make the assumption that evolution has evolved to optimise for certain properties of the reaction system, such as maximising biomass production, and simulate this maximisation to find rates; unbiased methods make no such assumptions.

Unbiased methods are typically analytic decompositions of the flux space such as Elementary Flux Modes [18], although some uniform Monte-Carlo approaches exist, while biased methods are typically variants on Flux Balance Analysis.

Section 2.8 describes in detail the field of available techniques for both biased and unbiased optimisation, in conjunction with figure 2.2, which organises the taxonomy of different techniques visually.

However, the central dichotomy of purpose, as well as methodology, between biased and unbiased methods is extremely important to understanding the purpose of this thesis: biased methods are primarily predictive, since they focus on specific model states, while unbiased methods aim to produce descriptive results. On a high level, these two ap-

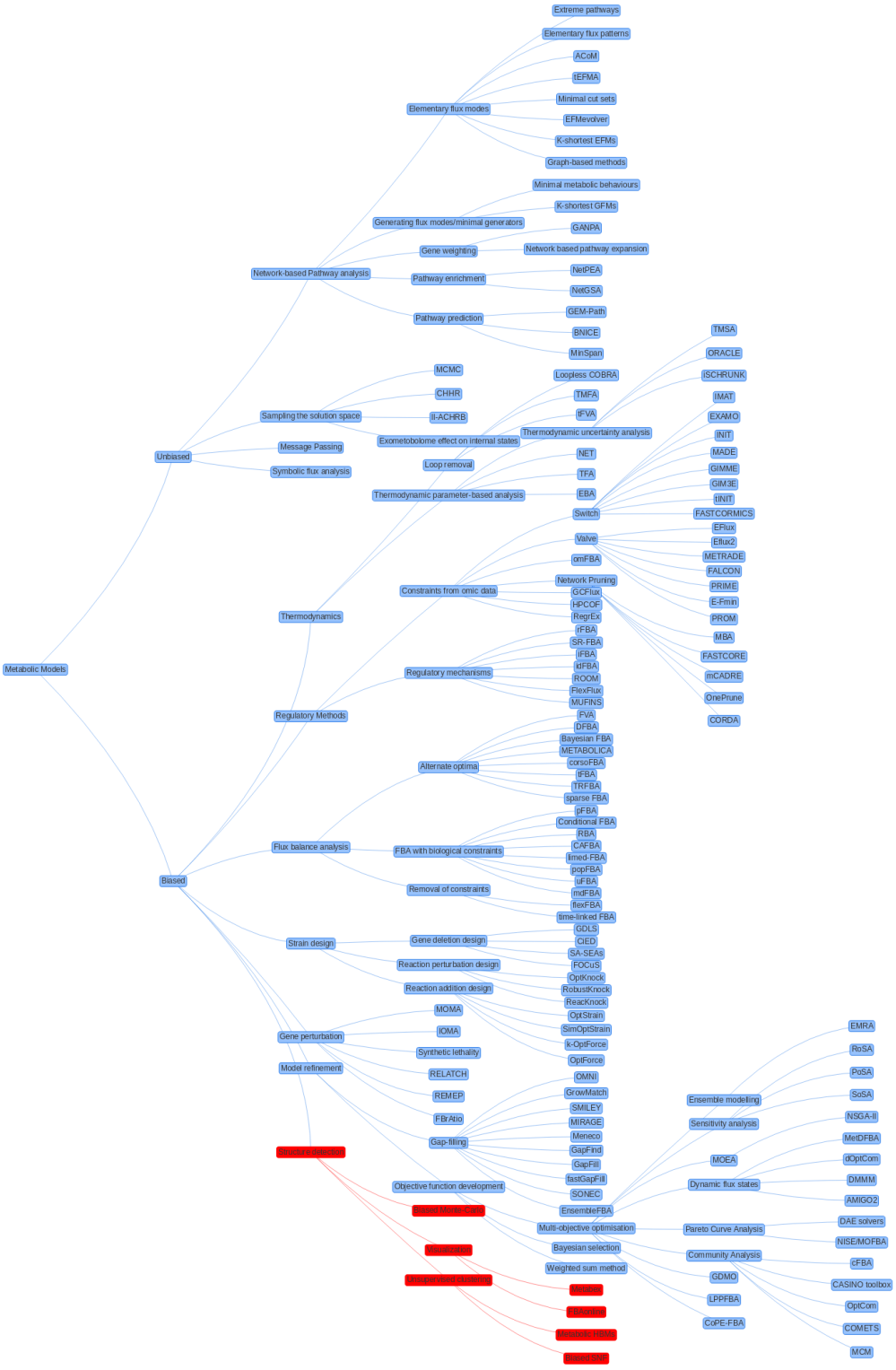


Figure 2.2: A tree diagram providing an overview of the field of metabolic modelling. Red nodes represent techniques and software introduced in this thesis: Biased structure detection techniques by combining Monte-Carlo simulations, visualisations and unsupervised clustering. A zoomable version is available online at `www.cl.cam.ac.uk/~mjc233/metabolic_modelling`.

proaches exist because whole cell metabolic models are very complex systems. Unbiased approaches such as Elementary Flux Modes are effective as descriptive tools, in that they capture uniformly all possible states of the model, but this is not necessarily possible with large models, nor is it always desirable, since some model states are more important than others. By contrast, biased models capture individual states of the system, one by one. This makes them very good at specific predictive tasks, but because they only capture one state at a time, they provide little descriptive power—they are very good at answering questions like 'what happens if this reaction stops?', but they are very poor at answering questions like 'which pathway is this reaction in?'. One purpose of this thesis is to combine these two approaches by using large scale biased simulations to describe metabolic networks. The aim is to provide descriptive tools over multiple states of models, but with the advantages of biased tools, by focusing on flux distributions that are optimal under some conditions.

2.6 Optimisation

Mathematical optimisation refers to searching the domain of a function in order to find an argument or set of arguments such that the result of the function, or some property of the result of the function, is as large or as small as possible. The most common objective is simply to minimise the function result, where this result is often a measure of error or cost.

In metabolic modelling it is common to see two different types of optimisation which serve quite different purposes, but are easily confused, so it is worth distinguishing between them.

The first is optimising the network fluxes to maximise the biomass objective function. Because of the assumptions made about the metabolic network, this optimisation problem can be expressed as a set of linear inequalities in what is termed canonical form. This structure can be exploited by techniques such as the Simplex algorithm to find the global optimum very quickly. We can term this part the inner optimisation problem. It is concerned with choosing flux assignments to simulate a plausible biological result.

Secondly, it is common to also use an outer optimisation algorithm, which alters the parameters to the inner optimisation problem. This outer optimisation is typically concerned with the constraints to reaction rates (rather than reaction rates themselves), and at each iteration will test a set of reaction constraints by using the inner optimisation algorithm to find the fluxes implied by those constraints.

Whilst the inner algorithm can typically be evaluated very quickly (typically less than 1 s in the applications described here), the outer optimisation algorithm cannot easily exploit any structure of the function that it is optimising over (which involves evaluation

of the inner algorithm), and so must run the inner algorithm at every iteration, meaning that a much more generic optimisation algorithm must be used, with much longer running times.

A discussion of linear programming (the typical technique for solving the inner optimisation problem), and evolutionary optimisation (a common technique for solving outer optimisation problems) follows.

2.6.1 Linear programming

Linear programming ('programming' is here used in the pre-computational sense of designing a schedule) is a technique for optimising a certain restricted class of systems which include metabolic networks. By virtue of their restricted problem class, linear programming solvers can be made much, much faster than more general optimisation techniques like gradient descent, whilst also providing much stronger guarantees about the result optimality.

Linear programming addresses optimisation problems that can be expressed in the form 'maximise $\mathbf{c}^T \cdot \mathbf{x}$, subject to linear constraints $\mathbf{x} \geq 0$ and $\mathbf{A} \cdot \mathbf{x} \geq b$'. These constraints can be interpreted as a polyhedron in which any valid solution must fall. The solution is the point on the polyhedron that is furthest in the objective direction. The simplex algorithm exploits the fact that a global optimum must be found at one of the vertices of this polyhedron, and works around the vertices to find this solution. Where an edge or face of the polyhedron is perpendicular to the objective direction, an infinite number of globally optimal solutions may exist, but there will still be one at each vertex adjacent to this edge or face.

Linear programming is used in a wide variety of contexts, but economically important resource allocation problems are often particularly amenable to this approach. Fortunately, this creates a relatively lucrative market that has produced a number of fast and highly optimised solvers, both open source and commercial. The three solvers used most in this research are Gurobi, one of the fastest commercial solvers, with an academic licence, GNU Linear Programming Toolkit (GLPK) [19–21], a relatively fast and reliable open source solver, and Embedded Conic Solver (ECOS) [22, 23] for some testing and simple problems, since it is self-contained and easy to install.

2.6.2 Evolutionary optimisation

Evolutionary optimisation is a description for a wide range of optimisation algorithms that generally maintain a population of independent solutions, generate new solutions from the old solutions via a random mutation process, and discard low quality solutions in favour of higher quality ones. Many evolutionary optimisation algorithms take inspiration from

features of biological evolution, such as generating new solutions by a pairwise sexual recombination process.

Evolutionary optimisation algorithms do not generally keep track of population level aggregate statistics, do not generally require a concept of landscape gradient, and are afforded some robustness against local minima by keeping a wide population. These properties make them suitable for use in optimising systems that are difficult to characterise. Furthermore, their wide population of solutions means that they can be used as part of a process of characterising a complex fitness landscape, since they can give information on a variety of local minima.

Both of these properties make them suitable for use on the outer optimisation step of metabolic networks, since they are able to address these complex problems whilst also providing information on the shape of the entire solution space.

2.7 Networks

The word 'network' generally denotes a set of nodes and edges, where each edge connects two nodes (or sometimes a node to itself), and each node has zero or more edges connected to it. This description is applicable to a vast number of areas, with different explicit assumptions about network structure and the properties of nodes and edges, and often implicit assumptions about what different features mean.

2.7.1 Metabolic networks

This thesis primarily discusses chemical reaction networks, which are a type of flow network. Comparing to other flow networks might lead us to jump to a description where metabolites (chemicals) are nodes, and reactions are edges, a bit like a logistics network where we could think about metabolites as stores of matter with conversions between them. Alternatively, we might think about reactions as nodes, with metabolites being edges between them, much like in a traffic network where junctions allow flow to change directions.

However, metabolites can take part in multiple reactions, and reactions can have more than one substrate and one product, so in fact both must be considered as nodes, edges between them purely indicating stoichiometric coefficients. A missing edge is a coefficient of 0. Since reactions can only be connected to metabolites and vice versa, this graph is bipartite. The alternative to this would be to use the hypergraph generalisation, where edges can connect multiple nodes.

2.7.2 Similarity and distance networks

When discussing machine learning it is also important to touch on the concepts of similarity and distance networks, which are networks where edge weights represent the similarity or dissimilarity between nodes, under some measure. When discussing the representation of these quantities as networks, as opposed to distance or similarity matrices, it is also necessary to consider the implications of unconnected nodes. When nodes are not connected many algorithms take this to be equivalent to an edge weight of 0. This means that for machine learning applications, similarity networks are often more convenient than distance networks, since it makes sense for unrelated nodes to have a similarity of 0, whereas in a distance network setting, unrelated nodes would have an undefined or infinite distance.

2.7.3 Network representations

There are a number of different ways to represent networks in a tabular or matrix format, with different relative merits in terms of their convenience for certain computational operations, and the mental model that they encourage. For metabolic modelling, there are a few representations that are particularly useful. Adjacency matrices are matrices where both rows and columns represent nodes, whilst non-zero entries in the matrix represent edges. These are convenient because, if we use the bipartite nature of the graph to only list metabolite nodes in rows, and reaction nodes in columns, then the adjacency matrix of the network is the stoichiometric matrix of the reaction system. If we take a sparse representation of the adjacency matrix, we get an edge list, which in the context of metabolic modelling is a list of metabolite, reaction pairs. Finally, a convenient format is a reaction list, which can be interpreted as a kind of adjacency list, where for each reaction node we have the name and stoichiometry of each metabolite that is involved.

Representing networks in a manner that is amenable to machine learning is an open area of research, with a number of approaches introduced recently [24–26]. Chapter 6 discusses an approach to breaking down a large network learning problem into a series of smaller ones that are more similar to traditional tabular datasets.

2.8 Review of state of the art in genome-scale metabolic modelling techniques

2.8.1 Constructing, obtaining and improving metabolic models

Whole genome metabolic models are available online in repositories such as the Kyoto Encyclopaedia of Gene and Genomes, also known as KEGG [4], the Biochemical Genetic

and Genomic knowledge-base (aka BiGG) [27], the BioCyc collection of pathway/genome databases [28] MetaNetX [29] and ModelSeed[30], among many others. In addition, many of the newest are available first only as supplementary information to papers. The preparation of a genome-scale metabolic model involves the reconstruction of all metabolic reactions taking place in the organism, annotated and extended with supplementary information on the genes, metabolites and pathways. Reconstructions vary significantly in quality, both in terms of their format and ease of parsing and the completeness of their underlying information, and as such, manual curation and gap filling is sometimes required [31]. One technique for this is to reconcile predictions from models with *in-vivo* findings in order to identify gaps in our knowledge of metabolism [32].

Inconsistencies often exist between models and experimental data. This can be false positives or false negatives for boolean model properties, or poor correlations in numeric model properties. Algorithms exist to support the identification and correction of some classes of inconsistencies, such as Grow Match [33], SMILEY [34] and Optimal Metabolic Network Identification (OMNI) [35]. Finding inconsistencies can not only lead to improvements in model quality, but can sometimes also guide basic biological research [36]. GapFind and GapFill are examples of optimisation procedures which identify problematic metabolites which are known to exist in a cell but for which no known reaction either produces or consumes them, and propose mechanisms to restore pathway connectivity for these metabolites [37]. FastGapFill [38] is an extension of FASTCORE which incorporates flux and stoichiometric consistency to guide the gap filling process. Metabolic Reconstruction via Functional Genomics (MIRAGE) [39] conducts gap-filling by integrating with functional genomics data to estimate the probability of including any given reaction from a universal database of putative gap-filling reactions in the reconstructed network. This enables selection of the set of reactions, the addition of which is most likely to result in a fully functional model when flux analysis is done again. Many models also integrate signalling and regulatory pathways with metabolic networks in order to add information regarding underlying mechanisms.

2.8.2 Unbiased methods

As described earlier, unbiased methods search the whole solution space of a metabolic model to find sets of statistically analysable states without requiring the definition of a specific objective function. These techniques are therefore similar in principle to other unsupervised techniques in machine learning, although they aim to explore the statistical properties of a network model, rather than the tabular datasets that are pervasive in general purpose unsupervised methods.

Network-based pathway analysis is a large family of unbiased methods which assess the main properties of biochemical pathways [40]. Gene Association Network-based Path-

way Analysis (GANPA) improves upon this process by adding gene weights to determine gene non-equivalence within pathways [41]. A similar approach to this was recently proposed to compare pathway significance by constructing weighted gene-gene interaction networks in healthy and cancerous tissue samples [42], which could then be used to expand pathways for each set of samples and to compare their topologies. Network-based pathways enrichment analysis is another technique to identify a greater number of gene interactions. An example of this is NetPEA, which utilizes a protein-protein interaction network in combination with a random walk to include information from high throughput networks and known pathways [43]. The NetGSA framework is another example of where condition-specific data has been used in combination with network estimation in order to improve the detection of differential activity in pathways [44].

A variety of different methods can be used for network-based pathway analysis, to calculate the set of routes through the reaction network and the corresponding matrices which represent their stoichiometry. Elementary flux modes (EFMs) describe the minimal, non-decomposable set of pathways that operate within a steady state system. These are found by the repeated removal of single reactions until a valid steady state flux distribution cannot be calculated [18]. This process tends to yield a vast number of common functional motifs - this is one of the problems common to unbiased methods which the techniques in this thesis avoid. However, other techniques exist to attempt to address these large numbers of motifs. Agglomeration of Common Motifs (ACoM) can be used to cluster these motifs, with overlap between classes allowed [45]. Another approach is to determine a single elementary flux mode by solving an optimisation problem using EFMevolver [46]; this can draw attention to significant elementary flux modes. Finally, K-shortest EFMs enumerates elementary flux modes in order of their number of reactions, since the shortest pathways are often of most biological interest due to their high flux and experimental tractability [47].

A further approach to network based pathway analysis is Generating Flux Modes [48]. The set of Generating Flux Modes is the smallest set required to define the geometry of the flux space using a null-space algorithm. The number of GFMs is often extremely large, but it is possible to identify specific subsets with more reasonable computational requirements [49]. In addition, there exist methods for dimensional reduction of flux cones, such as minimal metabolic behaviours [50] and minimal generators [51]. One method is to search for the shortest path between a given pair of end nodes [52], but the assumptions made in this approach are often criticised as overly simplistic, since they ignore actual reaction stoichiometry, throwing away a large amount of information [53]. Thermodynamic constraints can be used to add additional network information, such as in tEFMA [54], which removes thermodynamically infeasible EFMS using network-embedded thermodynamic (NET) analysis [55]. This use of thermodynamic information can be extended to

identify for physiologically significant EFMS, since the largest thermodynamically consistent sets (LTCSs) of EFMs often represent condition-specific metabolic capabilities [56].

Extreme pathways are an important subset of EFMs [40]. They consist of the minimal, independent set of reactions required to exist as a functional unit, and are characterised by a set of convex basis vectors which represent the edges of the steady-state solution space [57]. Most of the methods described here are intended to reduce dimensionality, however EFMs increase dimensionality, since they require that each reversible reaction is converted to forward and backward irreversible reactions [58]. Minimal cut sets (MCSs) are the set of EFMs which are required for a nonzero value of the objective function [59]. This is useful when identifying the set of single changes that would be required to induce some effect on the network. Elementary flux patterns (EFPs) define all potential elementary routes for steady state fluxes as set of indices; they can be mapped to EFMs in order to include pathway interdependencies [60]. A number of frameworks have appeared that provide the capability of combining some of these approaches to synthesis pathway prediction and scoring the best candidates [61], such as GEM-Path [62] and BINCE [63].

Monte-Carlo sampling, message passing [64], and symbolic flux analysis [65] are often incorporated into unbiased methods. This thesis includes an example of a Monte-Carlo approach which combines biased and unbiased aspects: biased towards a particular area of the flux space, but characterising the shape of the surrounding area in a way similar to many unbiased techniques. However, Monte-Carlo sampling is often also used in a fully unbiased manner, such as in [66], where Markov chain Monte-Carlo is used to generate uniform samples from genotype space. Various approaches to speeding up uniform sampling have also been suggested, such as sampling from an ellipsoid surrounding the solution space [67].

Of the various intracellular data types, metabolic data is among the most closely correlated with observed phenotype [68], which means that integrating data from observed phenotype data into a constraint based model can improve predictions of metabolic activity [69, 70]. The MetaboTools package is an example of a toolbox designed to support this kind of approach [68].

2.8.3 Biased methods

Biased methods are those which are biased towards a particular objective function, such as maximizing biomass output, with the metabolic network serving as a system of constraints on this objective function.

2.8.3.1 Flux Balance Analysis (FBA)

As described previously, the most popular and well characterised biased method is Flux Balance Analysis (FBA), and many other biased methods are variants of it.

A Flux Balance Analysis problem has three basic components:

- a set of reactions, expressed as their stoichiometric equations between metabolites,
- a set of lower and upper rate bounds on these reactions, and
- an objective function: a linear combination of reaction rates that must be maximized or minimized.

Flux Balance Analysis then consists of finding an assignment of reaction rates which maximizes the objective function while remaining consistent with the reaction equations and rate bounds. For a system with r reactions and m metabolites, these components can be defined as follows:

$$\mathbf{lb} \coloneqq \text{the vector of lower bounds, length } r, \tag{2.1}$$
$$\mathbf{ub} \coloneqq \text{the vector of upper bounds, length } r, \tag{2.2}$$
$$\mathbf{c} \coloneqq \text{the objective function vector, length } r, \tag{2.3}$$
$$\mathbf{A} \coloneqq \text{the stoichiometric matrix, with } r \text{ rows and } m \text{ columns}, \tag{2.4}$$
$$\mathbf{x} \coloneqq \text{the vector of reaction rates, length } r, \text{ which we aim to find.} \tag{2.5}$$

The optimisation problem can then be expressed as:

$$\begin{aligned} &\text{Maximise} && \mathbf{x} \cdot \mathbf{c}, && (2.6)\\ &\text{subject to} && \mathbf{lb} \leqq \mathbf{x} \leqq \mathbf{ub}, && (2.7)\\ &\text{and} && \mathbf{Ax} = 0 && (2.8) \end{aligned}$$

This is an example of a linear programming problem, which allows linear programming solvers to be used to solve FBA problems, as described in section 2.6.1. The ability to use linear programming for FBA makes it particularly attractive due to the speed of available linear programming solvers.

An extension to this is the use of a multi-level linear problem, where a problem is first maximized according to one objective ($\mathbf{c}$), and then subject to that maximal objective value, is maximised according to another objective ($\mathbf{c}'$), as follows:

$$\begin{aligned}
&\text{Maximise} && \mathbf{x} \cdot \mathbf{c}', && (2.9)\\
&\text{subject to} && \mathbf{lb} \leqq \mathbf{x} \leqq \mathbf{ub}, && (2.10)\\
&\text{and} && \mathbf{Ax} = 0, && (2.11)\\
&\text{and} && \text{Maximise} \quad \mathbf{x} \cdot \mathbf{c}, && (2.12)\\
& && \text{subject to} \quad \mathbf{lb} \leqq \mathbf{x} \leqq \mathbf{ub}, && (2.13)\\
& && \text{and} \quad \mathbf{Ax} = 0 && (2.14)
\end{aligned}$$

This can be easily implemented because $\mathbf{c}$ is typically only nonzero for a small number of reactions, so after the first optimisation round, $\mathbf{lb}$ and $\mathbf{ub}$ can simply be set to equal the value of $\mathbf{x}$ obtained for these reactions.

As described elsewhere, the most important biological assumption inherent in FBA is that the cell is in steady state, so the total production and consumption of each metabolite must balance, although certain pseudo-reactions can simulate uptake or excretion [71]. However, certain extensions exist to FBA, such as Dynamic FBA (DFBA), which approximate dynamic conditions as a series of steady states [72]. The results of such approaches can be validated using ^{13}C metabolic flux analysis [73]. Other dynamic approaches include dynamic multi-species metabolic modelling (DMMM), which models competition between species in a microbial community [74].

Unfortunately, Flux Balance Analysis models are sometimes underdetermined, so there is a need for approaches to try to improve the quality of the constraints in models [75]. Flux Variability Analysis [76] (FVA) is one such approach, which varies fluxes to find the maximum and minimum rates for a given biomass production rate [77]. Fast thermodynamically constrained flux variability analysis (tFVA) is a technique for speeding up FVA by removing a variety of thermodynamically infeasible reactions, as is described further in the next paragraph [78]. I take a similar approach in section 3.4.3.

Another way to introduce further constraints is via Thermodynamic metabolic flux analysis (TMFA) and thermodynamic variability analysis, which remove thermodynamically infeasible reactions and loops and generates information on feasible metabolite activity and free energy changes [79, 80]. Removing these loops avoids violating the loop law: there is no net flux around loops of chemical reactions in steady state [81] (this is equivalent to Kirchoff's voltage law). This technique of removing infeasible loops before solving a model can also be applied as a pre-processing step to most other approaches [82]. Energy Balance Analysis (EBA) is another approach to integrating thermodynamic constraints into FBA [83].

Other approaches to further constraining models focus on biological, rather than chemical, context. For instance, Parsimonious FBA (pFBA) is intended to identify the subset

of genes which maximise the growth rate in the model,and hence the stoichiometric efficiency [84]. Where flux rates are constrained by the availability of a related chemical, conditional FBA can be used to model this relationship, such as in *Elucidating temporal resource allocation and diurnal dynamics in phototrophic metabolism using conditional FBA* [85] and *Evaluating the stoichiometric and energetic constraints of cyanobacterial diurnal growth* [86]. Resource balance analysis (RBA) limits flux rates by protein availability, providing another constraint [87]. CAFBA, constrained allocation flux balance analysis, is another, more recent technique for modelling how protein availability affects metabolism, including multiple classes of protein [88]. Cost-reduced sub-optimal FBA (corsoFBA) combines protein and thermodynamic modelling components to constrain a model, reducing the growth rate below that that would be possible in a normal FBA model.

Bayesian flux estimation approaches [89] and the METABOLICA framework [90] attempt to address similar problems to in this thesis, although normally on much smaller sample sizes or models, and with a different sampling approach.

On the other hand, some methods relax some of the constraints of FBA, such as flexFBA [91], which relaxes the constraints around the biomass function in order to allow for improved predictions for cells that are not yet in steady state.

2.8.3.2 Regulatory methods

Regulatory methods for constraining FBA are those which incorporate information from cell regulation and related external data in order to make a flux balance analysis model more specific to a particular set of conditions. For instance, Steady-state Regulator Flux Balance Analysis [92] is used to measure the effect of constraints on metabolic genes, and Integrated FBA (iFBA) is capable of incorporating regulator and signalling pathways into an FBA model in addition to metabolism, and can incorporate multiple reaction speeds to give predictions of model dynamics [93, 94].

The simplest regulatory methods use a purely boolean on/off model of regulation, which obviously results in some information loss. Continuous information about regulation can be incorporated probabilistically, such as by PROM [95, 96] or QSSPN [97], or in discrete categories [53], but regulatory information is typically modelled as continuously controlling reaction rate, such as by FlexFlux [98] or the multi-formalism interaction network simulator (MUFINS) [99].

One of the most readily available sources of quantitative data to parametrise metabolic networks is gene expression data. An example of this is transcriptional FBA (tFBA), which uses continuous gene expression differences between pairs of real world conditions constrain FBA [100]. Transcriptionally regulated flux balance analysis (TRFBA) [101] is another recent method for converting gene expression data to flux bounds via a con-

stant multiplier, along with some ability to compensate for unavailable gene expression information.

There are a variety of sources for gene expression to use in metabolic network parametrisation. The most abundant but also most time consuming to collate is of course the supplementary information of various papers. More convenient databases also exist, such as the Gene Expression Atalas [102], Array Express [103], and the Gene Expression Omnibus [104]. Some of these, such as the Gene Expression Atlas, Gene Chaser [105] and Profile Chaser [106] also contain useful metadata [107].

Discrete 'Switch-based' methods for metabolic data integration Discrete, or 'switch-based' methods completely remove reactions which meet certain criteria - typically where the expression level for one or more genes required for the reaction is below a certain threshold [108]. A typical method that falls into this class is Gene Inactivity Moderated by Metabolism and Expression (GIMME) [109, 110], which performs standard FBA after removing reactions against a flat transcription threshold. The flux data obtained can then be compared with known active reactions as a validation step [111]. This can be extended to estimate turnover fluxes in the form of Gene Inactivation Moderated by Metabolism, Metabolomics and Expression (GIM3E) [112].

Discrete regulation methods are particularly appropriate for creating tissue specific models for multicellular organisms, since often certain pathways are largely dormant in certain tissues. This can be modelled by using mixed integer programming to fit discrete flux targets to the network based on associated gene expression data, and then using linear programming to fit the fluxes to these targets [113, 114]. This Integrative Metabolic Analysis Tool (iMAT) implements this procedure [115].

Another tissue specific discrete algorithm is Integrative Network Inference for Tissues (INIT) [116]. This aims to produce metabolic models which are specific to particular cell types by integrating protein abundance data—the steady state assumption is modified in order to better model protein production, and mixed integer programming is used once again to fit the model to experimental protein abundance data [117].

Continuous 'Valve-based' methods for metabolic data integration Continuous regulation techniques adjust the FBA input model smoothly, rather in discrete steps. Typically, they adjust the upper and lower bounds to reaction rates to constrain the model to match outside data, often by adjusting the bounds in proportion to some normalisation of the measured expression of genes associated with each reaction [108]. Examples of this include E-flux [118] and E-flux 2 [119], METRADE [120], FALCON [121] and PROM [95]. Some evidence indicates that this continuous model of the relationship between gene expression and protein concentration, and hence metabolic flux, is more biologically accurate than a discrete model [117, 122].

In the example of E-flux, the relationship between gene expression and reaction rate bounds is primarily based on the upper bound: when gene expression is low, the upper bound is low, and when gene expression is high, the upper bound is high (but the lower bound does not follow it up) [110, 118]. This models a situation of zero order kinetics, where the network flux as a whole is constrained by enzyme concentration bottlenecks. E-flux can generate underdetermined models without a unique solution, so E-flux 2 follows this with minimisations of the flux vector's Euclidean norm to fully determine the result [119].

Expression data-guided flux minimisation (E-Fmin) [123] is a method which fits fluxes to a function of gene expression level, much like GIMME, but in a continuous manner. FALCON estimates enzyme abundances from gene expression data explicitly, by contrast to most other methods which infer flux constraints directly from gene expression levels. METRADE combines gene expression data with multi-objective optimisation to investigate how phenotypes trade off between multiple potential biological objectives.

Network pruning Network pruning methods are similar to discrete data integration methods in that they turn off some reactions entirely, but they are much more aggressive, removing as many reactions as possible to fit some set of criteria. Examples include FASTCORE [124], FASTCORMICS [125], MBA [126], mCARDE [127] and OnePrune [128]. Owing to the combinatorial nature of testing reaction removals, these methods can be quite slow, but Cost Optimisation Reaction Dependency Assessment (CORDA) [129] offers a clever solution by assigning penalties to reactions implemented as a placeholder metabolite, and then optimising for the minimisation of this placeholder metabolite in order to identify which undesirable reactions are necessary for each desirable reaction. This notion of uncovering a smaller subnetwork of important reaction dependencies is similar to my aim in Chapter 6, although quantifying dependencies differently.

Other methods A number of methods take other approaches to creating biologically accurate models. For instance, the Regularised Context-Specific Model Extraction method (RegrEx) uses regularised least squares optimisation to fit fluxes to experimental data such as gene expressions [130]. The Huber penalty convex optimisation function (HPCOF) represents an alternative optimisation function which has produced results with good correlations to experimentally determined values, and with the removal of some difficult to measure parameters. Finally, other approaches exist which are designed to formulate objective functions, rather than relying on those determined experimentally [131].

2.8.4 Genetic perturbation

Although genetic perturbation methods share the same basic mechanics with regulatory methods, their aims are somewhat different. Whereas the regulatory methods described in section 2.8.3.2 are primarily intended to model natural biology as well as possible, genetic perturbation methods are primarily intended to perform simulations of genetic knockout experiments. This capability can guide the scientific process by providing information about what experiments are likely to be productive in pursuit of a specific goal. For instance this could be design of modifications for genetic engineering, where the goal is often high excretion of a specific metabolite, or it could be identification of synthetic lethal knockout [1] combinations at a rate far faster than could be achieved by *in vivo* experimentation [133].

A variety of methods exist for finding and examining synthetic lethal sets of genes. For a set of n genes, the number of potential synthetic lethal combinations of size k grows roughly as $\frac{n!}{k!(n-k)!}$; given that n is typically on the order of 1000, this means that the number of combinations to be checked quickly becomes unreasonably large even for modest values of k. For this reason, many techniques are focused on ways of reducing the search space. For instance, Fast-SL interprets gene-protein-reaction associations directly to reduce the problem size to a more manageable level [134]. Another technique is to use minimal cut sets to identify synthetic lethal reaction sets from the reaction network topology directly, and infer gene sets from these [135]. Data Mining Synthetic Lethality Identification Pipeline (DAISY) is a statistical approach to predicting synthetic lethality by combining genomic survival of the fittest, shRNA-based functional experiments and pairwise gene coexpression [136]. There are also methods for dosage lethality methods, such as Identifying Dosage Lethality Effects (IDLE), which simulates pairwise knockouts via gene expression levels. [137]

Another area of research in genetic perturbations to metabolic models is improving model assumptions to make better predictions about behaviour after perturbations.

For instance, Minimisation of Metabolic Adjustment (MOMA) is a technique where the most likely post-perturbation state is assumed to be that which is most similar to the pre-perturbation state [138]. Similarity is modelled here by Euclidean distance, which means that finding the new flux space becomes a quadratic programming problem. While these are slower to solve than linear programming problems, since the simplex algorithm is not applicable, there are nevertheless a wide range of mature tools available [139].

IOMA, short for Integrative 'Omics Metabolic Analysis [140], is another model which uses quadratic programming, allowing it to incorporate kinetics information to give more accurate predictions of post-perturbation behaviour.

[1] Synthetic lethality refers to the situation where a combination of mutations are lethal, even though the constituent mutations are not. [132]

By contrast, Regulatory On/Off Minimisation (ROOM) [141], does not minimise Euclidean distance from wild type, but instead uses mixed integer linear programming to minimise the number of fluxes that undergo large changes, using an objective function that includes change thresholds.

Flux Balance Analysis with Flux Ratios (FBrAtio) adds flux ratio constraints to a standard FBA model via modifications to the stoichiometric matrix [142, 143]. This helps with an underdetermined model, and allows a perturbation to be performed while enforcing that the behaviour of other reactions does not change.

The RELATCH [144] approach is another way in which a reference flux distribution can be used to improve predictions of behaviour under perturbation, in this case using a small number of hand chosen constraint parameters, representing different ways of reaction to perturbations.

2.8.5 How this work fits in

This thesis is about the creation of unsupervised techniques for knowledge discovery, rather than specific predictive tasks. This unsupervised goal that is currently primarily covered by unbiased techniques, which operate purely on network structure itself, and aim to quantify the possible behaviours of metabolic networks. Examples of this kind of approach include Elementary flux modes [18] and Generating flux modes [48], and Extreme Pathways [40]. Much of the progress on these techniques has been aimed at either constraining the solution spaces further, such in Gerstl *et al.* [54], or increasing the speed of existing techniques [145]. These techniques are concerned with characterising the space of possible flux distributions for a given metabolic network, and this is a goal that has also been attempted using Monte Carlo simulations, such as in [67] and [146].

However, these techniques make the assumption, whether explicitly or implicitly, that the behaviour of a cell is well modelled by a uniform distribution over the space of possible fluxes. One of the goals of this thesis is to explore how various biased sampling techniques, presented in Chapter 3, can improve on this, while still allowing the characterisation of a flux distribution, in what can be termed a partially biased approach. This not only has the benefit of being arguably more realistic, but also matching the techniques used in section 2.8.3.2, so that similar techniques can be used with or without experimental data. The most similar previous work on this area appears to have been [147], although this is with a sample size 10 times smaller, and no sophisticated post-processing.

Because this thesis uses techniques based on Flux balance analysis to perform this partially biased analysis, it is also worth comparing the techniques here to other related techniques, even when they are used in a different application. In view of the need to generate a set of varying metabolic models, the various conditional regulatory methods are most applicable here. Discrete regulatory methods are touched on in Chapter 3, which are

comparable to GIMME [109, 110], but I use continuous methods far more in this thesis, taking some inspiration from the E-fluxes [118, 119] for the design of both gene regulation strategies themselves, and the sampling strategies in Chapter 3, which required a similar approach of choosing flux limits in order to generate a particular flux distribution.

2.9 Conclusion

This chapter has covered relevant background material to provide context for the rest of the thesis. This started with general biological background, in section 2.2, and progressed to gradually more specific biological background, with section 2.5 and section 2.8 describing details of the most comparable related work. Finally section 2.6 and section 2.7 introduced those technical and implementation background details that are relevant to some or all of the following chapters.

Some background for algorithms described in only specific chapters is left for those chapters. The next chapter, Chapter 3, presents a taxonomy that is useful to understand the mechanism and purposes of the techniques described in this thesis.

CHAPTER 3

Characterising state spaces of flow networks: sampling metabolic models

3.1 Introduction

In Chapter 2 it was discussed how, under a number of reasonable assumptions, the metabolism of an organism can be modelled as a many dimensional polytope containing all feasible flux assignments. The two main groups of methods to analyse this polytope are unbiased methods, which attempt to decompose the flow network in order to achieve an understanding of every possible flux, and biased methods, which attempt to find the one true biologically correct solution.

However, there is a significant gap between biased and unbiased methods that is under investigated. Organisms are able to react to many situations, and so there is a large amount of merit in characterising the full range of phenotypes that they are capable of, but on the other hand we want to interpret these responses in a manner that is biased towards the most important and likely phenotypes, and we want to downplay detail that is less important. This more aggressive approach avoids the overly cautious responses given by fully unbiased methods which pin the correct answer down to somewhere in a few tens or hundreds of continuous dimensions.

The aim of this chapter is therefore to introduce a set of sampling techniques which fill this gap by finding sets of solutions which are all optimal, but for a set of models which are similar, but not identical in some way—for instance, different perturbations of a single model.

The most important property of these solution sets is that they exhibit variance that makes them suitable for characterising flow network properties during later analysis. The

success of this goal can primarily be evaluated through the success of the use of these datasets in the other chapters. In addition, particular attention is paid to attempting to mimic 'biologically realistic' variation. The only way of truly looking at biologically realistic variation is to base the sampling on real biological datasets, which is touched on in section 3.2.5, but outside the core scope of this chapter. However, even when the raw data is experimentally derived, transforming it into a usable model often requires extensive normalisation and data processing, and proving the correctness of this is often very difficult. For this reason the target of biological realism used in this chapter is mainly restricted to techniques which are biologically justifiable, in that they imitate a realistic biological mechanism—many of the techniques here have strong parallels with existing unbiased techniques discussed in Chapter 2.

The final goal of this chapter is to demonstrate that these techniques can actually be implemented to provide data for use in demonstrating techniques in the other chapters. Since metabolic models typically have large numbers of variables, the sample sizes required were very high, and by the nature of the use of random perturbation, many modifications of models were required. In addition, any kind of batch-based sampling requires that a large number of models is maintained and modified at any given time. These targets meant that the actual implementation of software capable of supporting the sampling discussed in this chapter was a significant part of the challenge. This new software enabled the creation of what I believe to be the two largest metabolic network datasets of their kind, at approximately half a million samples each.

3.1.1 Overview

In order to characterise phenotypes that imitate nature in a useful and justifiable way, we need to design a distribution from which to draw them. This requires some bias towards more biologically beneficial solutions, but also methods to induce variation, so that a larger space is explored.

Previous efforts [148, 149] to explore the flux space have sometimes taken an approach of blind, uniform Monte-Carol sampling: they pick a set of fluxes, check if it is feasible, and try again. This does have the effect of creating a very unbiased view of the shape of the flux space, but as described in section 2.4 there is no reason to expect a natural population to exhibit such a uniform distribution.

This implies that it in order to increase the realism of the sample, a superior approach is to use Flux Balance Analysis to find a solution that is on the surface of the flux polytope for a given set of constraints. This implies some degree of biological optimality, which is a reasonable assumption [150] and also has a performance advantage, since it allows us to avoid checking infeasible fluxes.

This chapter is therefore separated into three sections:

- Section 3.2 describes a selection of ways in which metabolic models can be modified to approximate real world variation. This defines the sampling space.
- Section 3.3 describes sampling strategies that can be used to define the draw distribution within the sampling space.
- Section 3.4 describes some concrete examples of pairings of techniques from section 3.2 and section 3.3.

Expressed mathmatically, the aim of this chapter is to sample vectors of reactions rates $\mathbf{x}$ which, or a system with r reactions and m metabolites, where

$$\mathbf{lb} \coloneqq \text{the vector of lower bounds, length } r, \tag{3.1}$$
$$\mathbf{ub} \coloneqq \text{the vector of upper bounds, length } r, \tag{3.2}$$
$$\mathbf{c} \coloneqq \text{the objective function vector, length } r, \tag{3.3}$$
$$\mathbf{A} \coloneqq \text{the stoichiometric matrix, with } r \text{ rows and } m \text{ columns}, \tag{3.4}$$
$$\tag{3.5}$$

satisfy the optimisation problem

$$\text{Maximise} \quad \mathbf{x} \cdot \mathbf{c}, \tag{3.6}$$
$$\text{subject to} \quad \mathbf{lb} \leqq \mathbf{x} \leqq \mathbf{ub}, \tag{3.7}$$
$$\text{and} \quad \mathbf{Ax} = 0 \tag{3.8}$$

as previously described in section 2.8.3.1.

Variation induction, in section 3.2, describes of methods of generating variance for this sampling, by exploiting the typically underdetermined nature of this optimisation problem, or by modifying the problem itself, either the upper and lower bounds, **lb** and **ub**, the objective function, **c**, or the network structure, **A**. Sampling priors, in section 3.3, describes the data sources that can be used to choose the values for these modifications, in a way that is as biologically justifiable as possible. This either means attempting to infer values from experimental sources, or simulating biological processes. Finally, the case studies in section 3.4 give specific instantiations of these sampling techniques, the results of which are used in later chapters.

3.2 Inducing variation to create sample spaces

Three obvious methods present themselves for inducing variation in a metabolic model in order to create a range of solutions: A, modifying the network structure itself, B, modifying the constraints to reaction rates, and C, modifying objective values.

Modifying network structure by removing reactions is equivalent to limiting reaction rates to zero. Similarly, from both a biological and technical standpoint, adding reactions is best modelled by adding a short list of reactions with rate limits of zero to the model, to create another instance of modifying rates. However, these modifications are worth breaking down as separate instances of constraint alterations, due to their different biological interpretations and technical lineage.

3.2.1 Modifying the environment

One of the most straightforward ways in which constraints can be varied is by varying the substrate available to the cell. In a computational instantiation, this means varying the constraints on the uptake proxy reactions to simulate different combinations of available resources. Obviously, this corresponds to the fact that multiple environments exist in the wild, however there is also a more subtle point here: wild organisms do not typically evolve for one exact set of conditions, instead they evolve to cover a niche. This means that rather than being optimised for one exact substrate, and 'hoping' that others will be similar enough that they survive, evolution is optimising for the sum of growth over the conditions experienced.

Model organisms grown in laboratories for many generations on tightly controlled media are in fact a very unusual situation, where the organism is optimised for one precise substrate, but then potentially introduced to something entirely new. It is worth noting that this optimisation occurs on an inter-generational timescale as well, in that a genotype or set of genotypes that can evolve rapidly to deal with the set of likely new situations is more successful than one that cannot.

Mathematically, this corresponds to modifying **lb** and **ub** in equation (3.6), specifically, those elements of **lb** and **ub** which correspond to auxiliary exchange pseudo-reactions. These are reactions which represent the introduction or removal of material from the system, and are hence one sided - rather than having both substrates and products like a real chemical reaction, they have just substrates or products, and act as a sink or source.

Section 3.4.4 provides an example of environment modification sampling.

3.2.2 Modifying network structure by removing reactions

Binary reaction knockouts, particularly by single gene deletions, is the archetypal practical application of metabolic network models, typically featuring as a 'hello world' style example in tutorials for metabolic network analysis packages. This is because it is simple conceptually and also relatively easy to test, since it mirrors *in vitro* single gene deletion studies.

In terms of technical implementation it is often easier to simulate by fixing both the

upper and lower bounds of a reaction at zero, since this avoids needing to actually alter the model structure. This corresponds to setting values of $\mathbf{lb}_i = \mathbf{ub}_i = 0$ for values of i corresponding to reactions that need to be removed. This is equivalent to, and simpler to implement, than completely removing reactions from the model, which would involve reducing the lengths of all the vectors in equation (3.6), and removing columns from $\mathbf{A}$. Although leaving redundant values in the model has a slight negative impact on the time to solve the linear programming problem, most solvers identify these redundancies quickly meaning that the time saved would be much less than that required to modify all the problem variables.

3.2.3 Modifying network structure by adding reactions

Modifying metabolic models by adding new reactions is typically modelled by taking a path of adding putative new reactions with zero flux, and adjusting their fluxes to see their effects. There are four separate scenarios that we can seek to model by adding reactions.

Firstly, from a genetic design perspective, this can be used as a selection procedure to investigate the merits of engineering in new reactions. In this case, the set of putative reactions is a list of those which could plausibly be engineered in, such as from a related species with particularly interesting properties.

Secondly, we could be attempting to expand an existing metabolic model, by checking for reactions which might be missing. These are normally implied by the known existence of reactions or phenotypes which are pointless or implausible without them. For instance, if an organism is known to be able to consume a certain feedstock, and part of a known pathway for this exists, then this provides strong evidence that any missing reactions in that pathway are probably present as well. For this, the set of putative reactions is best provided by a closely related but larger metabolic model.

Thirdly, we could be attempting to model the natural variation in a population that is capable of horizontal gene transfer. In this case, we once again have a well known set of possible reactions controlled by genes mobilised by plasmids or other vectors. However, here we ideally want to assign a cost to having a reaction at all, since horizontal gene transfer is hypothesised to be a method to avoid all cells needing to replicate rarely used genes. This is something of a computational difficulty, because this kind of binary penalty pushes the model out of what is solvable using linear programming.

In each of these cases, much as in reaction removal, the best computation approach is to start with a larger model with various reactions i turned off via setting $\mathbf{lb_i} = \mathbf{ub_i} = \mathbf{0}$, as opposed to modifying the network structure.

Lastly, we might wish to model the much rarer event of evolution of new reactions, rather than simply moving existing reactions around. This is more conceptually complex,

and requires a model of the underlying mutation mechanism, which is itself a relatively poorly characterised field.

In *Iterative Multi Level Calibration of Metabolic Networks* [10], I experimented with the first and second scenarios, testing synthetic reaction additions and identifying missing reactions. This was successful at proposing synthetic reactions to transplant from an *E. coli* model, but with the species that were tested, missing reaction detection was less effective, possibly due to the fact that one model was already derived from the other.

3.2.4 Modifying objective values

Modifying the primary biological objective values shares many biological properties with modifying the environment. In one case, the materials available are changing as availability changes, whilst in the other, the materials required are changing as the exact challenges that the cell faces changes. In either case, we create a situation where we acknowledge that a phenotype is tailored as a compromise not just between multiple static objectives, but between the range of values that those objectives can take. A starting approximation to this is the use of a small number of discrete objective functions included in metabolic models. The most common of these are:

- the ATP maintenance allowance, which is the raw energy required to survive;
- the biomass objective, which is the bill of materials required for growth; and
- sometimes a maintenance and repair function, which is the slightly different set of materials required for maintaining existing biomass.

These are typically used in the pattern of setting a limit on all but one and optimising for the last, but it is also common to explore the tradeoffs between these objectives.

There is also a separate, less biologically meaningful, but nevertheless very technically important application to modifying objective values: tie breaking. Section 3.2.6 describes how, for many models, maximising biomass does not fully determine the flux space, leaving multiple different solutions still feasible. Fixing the biomass objective at a maximum and then creating further objectives provides a method to break this tie whilst exploring possibilities further. These secondary objective functions should be biochemically reasonable, so it makes sense for them all to reduce fluxes in order to avoid futile loops and create a parsimonious result, but the ratios between these objectives can be altered in order to explore the multitude of possible ways to achieve the objective value.

Unsurprisingly, modifying objective values in equation (3.6) just involves modifying $\mathbf{c}$. However, if we take the approach of using objective function modification to perform tie breaking, this in fact necessitates a further round of optimisation with new values for $\mathbf{lb}$, $\mathbf{ub}$ and $\mathbf{c}$, denoted $\mathbf{lb}'$, $\mathbf{ub}'$ and $\mathbf{c}'$:

$$\mathbf{lb}'_i \coloneqq \begin{cases} \mathbf{x}_i, & \text{where } \mathbf{c}_i \neq 0, \\ \mathbf{lb}_i, & \text{where } \mathbf{c}_i = 0 \text{ and } \mathbf{lb}_i \geq 0, \\ \mathbf{lb}_i, & \text{where } \mathbf{c}_i = 0 \text{ and } \mathbf{lb}_i < 0 \text{ and } \mathbf{x}_i < 0 \text{ or} \\ 0, & \text{where } \mathbf{c}_i = 0 \text{ and } \mathbf{lb}_i < 0 \text{ and } \mathbf{x}_i \geq 0 \end{cases} \tag{3.9}$$

$$\mathbf{ub}'_i \coloneqq \begin{cases} \mathbf{x}_i, & \text{where } \mathbf{c}_i \neq 0, \\ \mathbf{ub}_i, & \text{where } \mathbf{c}_i = 0 \text{ and } \mathbf{ub}_i \leq 0, \\ \mathbf{ub}_i, & \text{where } \mathbf{c}_i = 0 \text{ and } \mathbf{ub}_i > 0 \text{ and } \mathbf{x}_i > 0 \text{ or} \\ 0, & \text{where } \mathbf{c}_i = 0 \text{ and } \mathbf{ub}_i > 0 \text{ and } \mathbf{x}_i \leq 0 \end{cases} \tag{3.10}$$

$$\mathbf{c}'_i \coloneqq \begin{cases} 1, & \text{where } \mathbf{x}_i < 1, \\ -1, & \text{where } \mathbf{x}_i > 1 \text{ or} \\ 0, & \text{where } \mathbf{x}_i = 0 \end{cases} \tag{3.11}$$

This is ensures that in the second round of optimisation, either $\mathbf{c}'_i \neq 0$ or $\mathbf{ub}'_i = \mathbf{lb}'_i = 0$ for all values of i, and optimises for small fluxes where possible and consistent with the previous maximum. This is the second round optimisation method implemented as a default in my Fbar package (section 3.4.1).

3.2.5 Modifying reaction rates directly using gene expression data

Gathering large amounts of real data about reaction rates is difficult, especially for internal fluxes. However, gathering large gene expression data sets is relatively easy, and so there is a natural draw to attempt to infer reaction rates from gene expression. This has been approached many times [95, 118–121], but the relationship between gene expression and reaction rate is in general unknown, and an area of active research with good evidence that the relationship is complex, as discussed in [7].

One might assume that the lack of a simple relationship between gene expression and flux might imply a fundamental problem with techniques that assume that flux can be predicted via a simple relationship with gene expression. However, in fact this is not the case, because these methods work on constraining the limits of reaction flux. This means that what they are actually achieving is a softened on/off regulation, which has much stronger biological support: if gene expression is sufficiently low, the enzyme concentration can be too low for the reaction to occur at all.

Although we do not know the exact shape of the relationship between a given gene expression and reaction rate, we would still like to exploit this common and high fidelity

data type to make predictions of higher level functionality. The current state of the art in interpreting gene expressions is ontology analysis, which uses black box correlations between genes and general areas of biological functionality, so any intermediate layer that provides a degree of interpretation and justification to inferences from gene expression data sets is welcome. This technique must use a strong prior from known properties of the metabolism, strong enough to ignore bad signals that are present in data sets with small sample sizes.

Completely controlling reaction rates directly from gene expression is not compatible in the FBA framework described in equation (3.6), because if $\mathbf{x}$ was fully determined by gene expression data, then there would be no optimisation step left to be done. However, most models contain large numbers of reactions which are not gene expression controlled, either because they are biologically spontaneous, or because the controlling gene-protein-reaction mapping is not included in the model. Moreover, control of reaction rates is typically not implemented directly, because for more than a handful of exactly chosen fluxes, it is not possible to find a solution which is consistent with all of them unless they are precisely correct, which is far beyond the capabilities of the current state of the art in both transcriptomics and metabolic modelling.

For these reasons, it is typically the case that gene expression values are used to generate relative flux constraints, which are implemented via adjusting **lb** and **ub**.

As described in section 2.8.3.2, a large number of different functions have been used for mapping gene expression levels to metabolic network constraints, from simple boolean switches to much more complex multi-step processes such as in E-flux 2 and SPOT [119].

However, the pre-processing and normalisation of gene expression levels is an important precursor to any regulatory algorithm, and one that is potentially overlooked, or addressed only implicitly by the flexibility of the regulatory algorithm itself.

For instance, in the Colombos dataset [151] used in Chapter 5 and Chapter 6, several stages of normalisation have already been done. This makes the dataset convenient and easy to use, but has the potential to lead to loss of information. This is unfortunately necessary because, due to its nature as an integrative project, the various different data sources for Colombos have not necessarily had the same postprocessing treatment already, and so aggressive normalisation is required to make them at least somewhat comparable.

By way of comparison, in section 4.2, the smaller amount of much more raw data required a customised normalisation protocol in order to convert the dataset to a suitable format, as described in section 4.2.1.

3.2.6 Taking advantage of slack in model

Large metabolic models are rarely fully determined. This means that most of the time, there will be more than one flux distribution which achieves optimality in their objective.

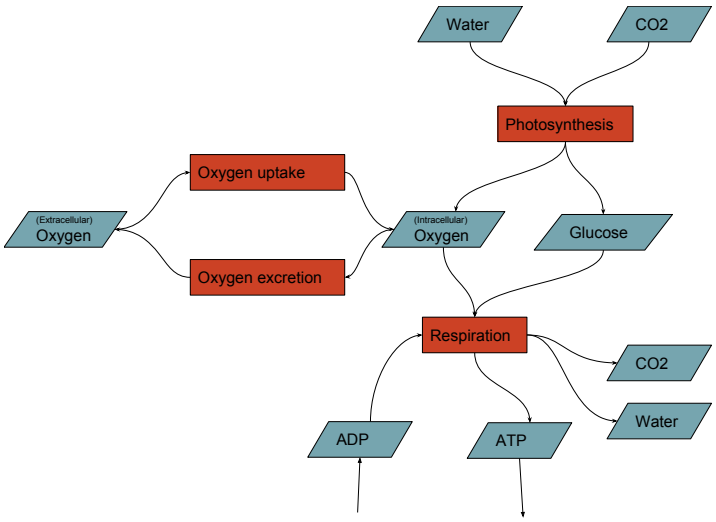


Figure 3.1: A simplified sketch demonstrating slack in a metabolic network. Assuming that the network is optimised for ATP production, and limited by light availability, there is slack in this model. This is because for any given value of ATP production, the cell must have been excreting and ingesting oxygen at the same rate, but that rate is not known.

This is referred to as *slack* in the variables that are not fully determined, and in many situations represents a fault with a model (see figure 3.1). Equation (3.9) describes one way to address this issue. However, in this application slack is in fact useful, because it means that a set of constraints to the metabolic network will not produce a single optimal solution, but a set which can be explored.

This means that it is possible to find some model variation without actually altering the model at all, but simply by sampling from the (potentially small) set of optimal solutions. In the terminology of equation (3.6), this would involve sampling reaction rates $\mathbf{x}'$ by varying $\mathbf{c}'$as follows:

$$\begin{aligned}
&\text{Maximise} && \mathbf{x}' \cdot \mathbf{c}', && (3.12)\\
&\text{subject to} && \mathbf{lb}' \leqq \mathbf{x}' \leqq \mathbf{ub}', && (3.13)\\
&\text{and} && \mathbf{A}\mathbf{x}' = 0, && (3.14)\\
& && && (3.15)
\end{aligned}$$

where

$$\forall_{i=1}^{\|\mathbf{c}'\|}\mathbf{c}'_i \sim \mathcal{N}(0,1) \tag{3.16}$$

$$\mathbf{lb}'_i = \begin{cases} \mathbf{lb}_i, & \text{where } \mathbf{c} = 0, \\ \mathbf{x}_i, & \text{where } \mathbf{c} \neq 0, \end{cases} \tag{3.17}$$

$$\mathbf{ub}'_i = \begin{cases} \mathbf{ub}_i, & \text{where } \mathbf{c} = 0, \\ \mathbf{x}_i, & \text{where } \mathbf{c} \neq 0, \end{cases} \tag{3.18}$$

$$\tag{3.19}$$

subject to

$$\begin{aligned} &\text{Maximise} && \mathbf{x} \cdot \mathbf{c}, && (3.20)\\ &\text{subject to} && \mathbf{lb} \leqq \mathbf{x} \leqq \mathbf{ub}, && (3.21)\\ &\text{and} && \mathbf{Ax} = 0 && (3.22) \end{aligned}$$

A normal distribution is used for $\mathbf{c}'$ here for simplicity; any continuous distribution is a reasonable choice. A continuous distribution is preferred to a discrete one because in this context it guarantees that $\mathbf{x}'$ is almost surely determined.

In the research described in section 4.2, I used this technique to identify and highlight fluxes that were most strongly affected by a given set of gene expressions.

3.3 Sources for prior distributions over sample spaces

When we lack a prior belief on the best sampling strategy to use with a variation induction technique, it is reasonable to devise the most 'fair' sampling method possible. However, we would ideally want to do better than this, and find a technique that produces a more realistic solution.

One of the most obvious and best sources for how to induce variation is to find a real world experimental source, so that we are selecting uniformly from a set of known real conditions. This is particularly relevant to environmental and gene expression based variation induction, because many datasets do exist for this [102, 105, 106]. However, real world sources are not always available, not always appropriate to every situation, and even where they do exist are not perfect information sources. In particular experimental sources have many potential sources of error from experimental and analytical steps, during which often make implicit assumptions about the expected shape of the distribution of values, and even if we had a perfect measurement of gene expression, this is only a weak to moderate correlate of protein abundance, let alone reaction rate [152, 153]

Furthermore, experimental data almost always measures collections of cells, meaning that they often do not capture variance within populations well.

Another option is via an evolutionary approach. We normally think of evolution as a technique for optimisation, which we use in situations where we want a single optimal solution. However, it inherently works on populations of solutions, which means that it is also suitable in this context for adjusting a population to optimally fit a landscape such that the typical individual in the population has high utility. This is clearly relevant to evolvable properties, such as in adjusting gene expressions or adding and removing reactions, because it simulates the real life sampling procedure. However, in addition it can act as a simulation of environmental variation when combined with these techniques, because it can pick up on the effects of simultaneous change in these aspects: individuals that require a large simultaneous change in both phenotype and environment are less likely than those that require only change in one at a time.

Modifying objective values is a more difficult area when it comes to choosing the pattern of variation induction. This is because the real mechanisms depend on complex and difficult to measure biochemical interactions. However, objective functions are typically based on experimentally determined biomass composition, leaving less room for variation.

3.3.1 Gene expression sampling

Gene expression variation induction can be used to generate populations of metabolic models in one of two ways: by computing flux constraints from a real world quantitatively measured gene expression dataset, or by using a gene-protein reaction mapping in conjunction with a different sampling approach, such as a uniform or evolutionary sampling approach.

In the research projects described in Chapter 5 we used the first approach of using a logarithmic function to map from each gene expression to a flux value, as introduced in [120]. If we define:

$$y_i := \text{the gene expression of gene } i, \tag{3.23}$$

$$\text{sgn}(x) := \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{3.24}$$

$$h(y_i) := \text{the flux adjustment from gene } i, \tag{3.25}$$

then the function used is

$$h(y_i) = (1 + |\log(y_i)|)^{\text{sgn}(y_i - 1)}, \tag{3.26}$$

The data was pre-normalised, and had a large number of samples allowing for a reasonable precision in the standard deviation measurement.

However, in section 4.2, the number of gene expressions available was much smaller (11 samples from 4 gene expression conditions), since all gene expressions came from the same series of experiments. This had the benefit that direct comparison was more valid, but meant that it was desirable to find a near optimal parameter assignment for the gene expression function itself and for the preprocessing of the gene expression vectors, in order to improve the signal to noise ratio in the resulting simulated reaction rates.

3.3.2 Evolutionary sampling

In *Iterative Multi Level Calibration of Metabolic Networks* [10], described in section 4.3, an evolutionary sampling approach was taken, specifically a multi-objective evolutionary approach using a gene expression function so that mutation could be conducted in genotype space, but evaluation could be conducted in metabolic flux space. This approach has the major benefit that it produces a population that describes a tradeoff between metabolic objectives, whilst providing a bias towards individuals that are evolutionarily accessible, in other words, those genotypes that can be obtained via a series of intermediates with at least moderately high fitness levels.

Section 3.4 also contains an example of an evolutionary sampling approach.

However, taking an evolutionary sampling approach has a large drawback in terms of the number of parameters involved. Evolution is an extremely complex process and many relevant variables, particularly mutation rates, are not known or easily measured. This leaves a situation where uniformity assumptions about mutation rates [154] have to be made which are difficult to justify.

3.3.3 Environment modification

As described in section 3.2.1, the situation of organisms growing in a stable, homogeneous medium is far more common in the laboratory than in the real world, and often less common in the laboratories than microbiologists would like. For the examples in this thesis, one goal of this chapter is to provide a data set that mimics real world variation in organisms. In order to do that, it is desirable to have a source of external variation in addition to internal genetic variation and environment modification is a way to provide that. Many studies of condition variation exist but they tend to be small and transcribing them to usable notation is a slow and difficult manual task, so it is much better to simulate these values.

Unfortunately, unlike with gene expression values, we cannot simply create new environments via a Gaussian random walk, because without feedback from the fitness of

the organisms, the environment could wander into an area where no models are viable. This happens when the limits on uptake from the environment are too strict for the linear programming solver to find a solution. Specifically, this only happens in models that include maintenance thresholds, minimum limits on biomass generation, ATP expenditure, and possibly other processes, which represent minimum biological upkeep requirements and preclude a zero-flux model. Instead, the environment must be constrained to an area where there exist viable models, whilst still varying enough to promote genetic heterogeneity. This idea of the organism fitness affecting the environment might seem backwards, but in reality this is just a matter of deciding to gather data only where it exists.

The case study described in section 3.4.4 provides an example of this.

3.4 Case studies

This section describes three case studies showing instantiations of the concepts described earlier in this chapter. The datasets yielded by these case study sampling strategies are used as demonstrations in the final two chapters, and this intention guided their creation. The desirable properties of these datasets primarily involved having reasonable amounts of variance, in order to cover a useful area of the solution space, and for inference algorithms to work well one them.

Firstly, and least interestingly, this meant that the values in the resulting metabolic models had to vary. This is in many ways one of the central tenets of this thesis: cells naturally gravitate towards a neighbourhood of advantageous metabolic states, but at the same time it is worth modelling their variance.

Secondly, this variance should have a stochastic component. The alternative, which would probably look like a kind of grid sampling, has multiple weaknesses. Not only is it more difficult to justify biologically, but it also does not scale well and is more difficult to implement because simulations must depend on each other. Furthermore, it runs the risk of inference algorithms picking up structure from the sampling strategy, rather than latent structure in the metabolic model itself.

Thirdly, the variance should be spread across as many reactions as possible. This requirement was the focus of section 3.4.3, where I tried to make sure that reaction rate limits were quite 'tight', such that many of the reactions had random constraints on them, to induce variance. The intention of this is firstly to make it possible to draw inferences about the behaviour of more reactions, since we do not learn much about reactions that do not vary much. This also has merit from a biological realism perspective, since it is generally assumed that most reactions in a cell are at least sometimes active.

Fourth, trivial correlations are to be avoided. This particularly concerns situations where multiple simulations yield results that are similar or identical up to a scaling factor.

This would be bad because such simulations, while not resulting in exactly the same numbers, do not yield further information on the actual relationships between reaction speeds. The use of stochastic constraints is a significant step to avoid this potential problem, when used in conjunction with making sure that limits were generally tightly linked to reaction rates. On examination of the datasets, it also appeared to sometimes be a potential issue during an initial 'burn in' period, where the network was limited by a single reaction. Throwing away early samples that were affected by this was the simplest solution to this.

Finally, and most elusively, there was the desire for the sampling strategy to be 'biologically justifiable'. This is a difficult concept to really justify, but it is part of the reason for taking an evolutionary environment modification strategy in section 3.4.4. The advantage of this is that it at least has its basis in solid theory: environments change, and organisms evolve to adapt to those changes. Of course, the exact details of how their environments typically change is a significant experimental question that is beyond the scope of this thesis, and the question of how exactly they tend to evolve given those changing environments is another step further in difficulty.

3.4.1 Implementation: Fbar

In order to implement the sampling strategies described here, as well as support the analysis described in the other chapters, it was necessary to create an R package, which I named *Fbar*, for Flux Balance Analysis in R [155]. This package provides facilities for flux balance analysis, but it takes a markedly different approach compared with previous tools. Most tools for flux balance analysis are targeted towards relatively interactive, exploratory use, with a large number of functions for activities such as modifying reactions one by one, and often contain complex previously published analysis processes as complete, first order operations. This is extremely convenient for new users to quickly get some results, and is designed to support a pattern of use where models are created, modified in a number of ways, and evaluated. However, this design makes these packages less useful for the goal of large scale simulations, where large numbers of models need to be created and then immediately evaluated.

In contrast to previous packages, Fbar is designed to support large scale simulations. Whereas in previous packages a typical work flow consists of loading one model, and then iterating cycles of modification and re-evaluation, Fbar works by loading a single base model, and then combining it with multiple parameters (typically flux bounds or objective vectors), to generate a set of models which can then be evaluated quickly, and potentially in parallel.

These changes reduce Fbar's overhead on top of the actual linear programming code to a negligible level, and also reduce load times by about 3 fold compared to the fastest

tested alternative, the Matlab package COBRA. In particular, for a large model, Fbar was able to achieve a model loading and parsing time of 0.05 s to 0.085 s, compared to over 1 s in the Sybil R package. This represents a time saving of 127 h for a 500 000 sample dataset, or over 5 days. Fbar also has substantial user friendliness advantages compared to alternatives: at the time of writing it is the only R flux balance analysis package which works out of the box.

It is available on Github at `https://github.com/maxconway/fbar` and is published on the Common R Archive Network (`https://cran.r-project.org/package=fbar`), where it is seeing significant levels of use—over 460 downloads per month at the time of writing.

3.4.2 Case study 0: Placeholder sampling approach

My first sampling approach was a simple one, designed to quickly produce small sample datasets, for use in testing the functionality of the interpretation functions and the supporting framework. It had to be simple to verify by eye but did not need to be especially fast, since it would only be used to produce small numbers of samples over small networks. However, it was necessary to take an approach that would produce variance immediately, without requiring any burn-in period.

The approach selected was to perform a first simulation of the network with relaxed, default constraints (giving a flux vector $\mathbf{x}$), and then to set the new lower ($\mathbf{lb}'$) and upper ($\mathbf{ub}'$) bounds as follows:

$$\mathbf{lb}' = \begin{cases} 0, & \mathbf{x} \geq 0, \\ \mathbf{x} * (1 - \mathcal{U}(0,1)^5), & \mathbf{x} < 0 \end{cases} \tag{3.27}$$

$$\mathbf{ub}' = \begin{cases} 0, & \mathbf{x} \leq 0, \\ \mathbf{x} * (1 - \mathcal{U}(0,1)^5), & \mathbf{x} > 0 \end{cases} \tag{3.28}$$

where $\mathcal{U}(a,b)$ is a uniform random variable in the interval $[a,b]$.

The term $1 - \mathcal{U}(0,1)^5$ has a distribution function as shown in figure 3.2, which constrains each flux by a small, randomly chosen amount, with a mean of 16.7%, and a median of 3.0%.

The total sample size used here was 1000.

3.4.3 Case study 1: Batch based proportional adjustment

The goals of my second sampling approach were to ensure that variance was induced in as many of the reactions as possible, and to ensure that performance was high enough to

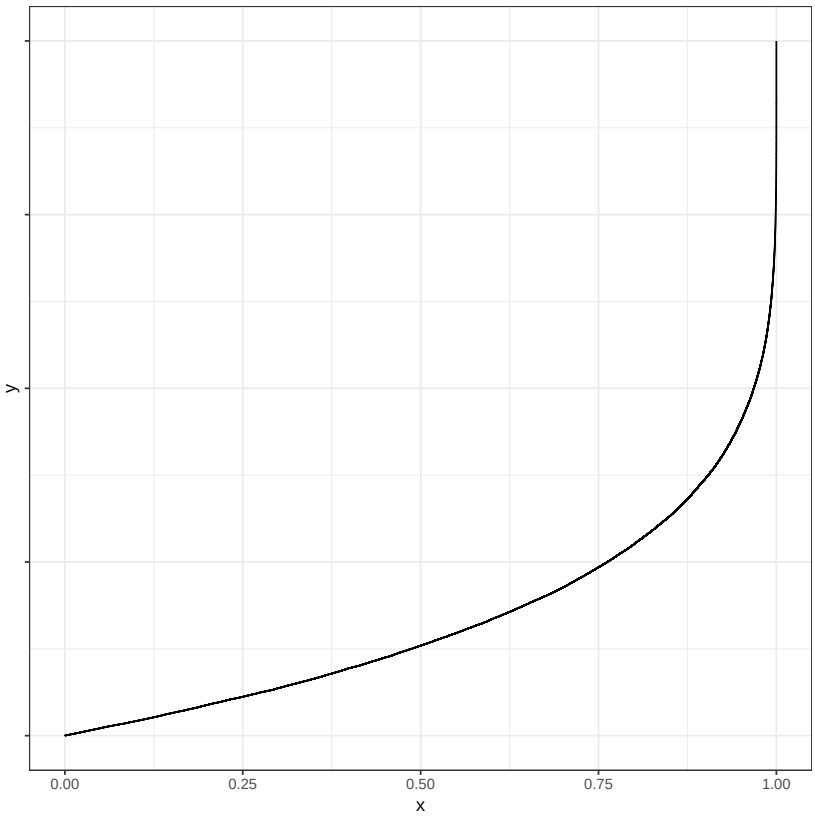


Figure 3.2: Empirical cumulative distribution function of the random constraint used in sampling approach 0.

create a large dataset in a reasonable time, even when simulating over a large network.

In order to achieve the required performance and create a code base suitable for long running simulations, I committed to using in place model adjustment rather than parsing the model repeatedly, since this gave an approximately 10 fold increase in speed at the sacrifice of only a small amount of flexibility around reaction addition and removal sampling. I also moved to a batch process, with 100 simulations per batch in this instance. Batches were saved in the high performance Feather file format, giving a 14 fold I/O speed increase compared with CSV.

The primary goal with this sampling approach was to ensure that as many reactions as possible showed variance, which meant that it was necessary for the reaction rate bounds to vary a small amount, near to the optimal rate of the reaction, given the other bounds. This is because if any reaction is too constrained, there is no feasible solution to the model, and it is non-viable. This can occur if the added constraints have too wide a variance (since then there is a high chance of an extreme constraint in at least one reaction), or if the constraints are not centred around the pre-constraint optimal values of the reactions, since then the composition of these small effects will make the model non-viable.

To find appropriate constraints, the constraint modification approach was split into two components, taking advantage of the batch structure.

Firstly, within each batch, upper and lower bounds for individual models were chosen from a random uniform distribution between 90% and 100% of the widest bounds for the whole batch (see equation (3.30)). In comparison with the approach taken in section 3.4.2, this removed the long tail of the distribution which could lead to a single constraint dominating the network.

$$\mathbf{lb}_{i,t+1} = \mathcal{U}(0.9, 1) * \bigwedge_{i=1}^{n} \mathbf{lb}_{i,t} \tag{3.29}$$

$$\mathbf{ub}_{i,t+1} = \mathcal{U}(0.9, 1) * \bigvee_{i=1}^{n} \mathbf{ub}_{i,t} \tag{3.30}$$

where $\mathbf{lb}_{i,t}$ and $\mathbf{ub}_{i,t}$ are the lower and upper bounds respectively for model i in batch t, which has size n. $\wedge$ and $\vee$ represent the minimum and maximum functions respectively.

Secondly, it was important to ensure that the whole batch bounds were close to the unbounded flux values (specifically within 111 %, given the previous section). To achieve this the bounds on each reaction were set to 1 % above the maximum flux value found in any model in the previous batch, unless this maximum flux value was very small (<1 % of the previous upper bound). If the maximum flux value was indeed very small, the bound was instead tightened by 1 %. This avoided constraining unused reactions too quickly, so that they could become active later in the simulation run, once the bounds on the used reactions had been tightened.

In this approach, at each iteration, each reaction is restricted to on average 5 % below the batch bound, and then the batch bound is only increased by 1 % above the previously found maximum flux. At a glance this can give the impression that the bounds would inexorably tighten by a net of 4 % at each iteration, but it should be noted that the batch bounds are set to the *maximum* of the fluxes in the whole batch, so that with the chosen batch size of 100, the bounds on an otherwise unconstrained reaction would actually move up by on average 0.899 % per iteration.

This simulation was continued for 5000 generations, with a batch size of 100, giving a total sample size of 500 000.

3.4.4 Case study 2: Evolutionary sampling with environment variation

As stated previously, evolution as existing in nature has the objective of maximising expected fitness over a population of solutions, in contrast to most designed optimisation algorithms, which are evaluated on the single best result that they produce. This is partly because natural evolution typically exists in a changing environment, which means that

over-specialisation can be detrimental.

For this reason, this case study attempts to imitate aspects of an evolutionary approach that create a 'balanced' population of solutions. The basic features of an evolutionary algorithm are operations for mutation, evaluation, and selection [156], so it makes sense to describe the sampling approach in terms of these operations.

Mutation was achieved via a Gaussian random walk in genotype space. Specifically, the value for each gene $g_i + 1$ in a model in generation i was sampled from a Gaussian, with standard deviation 0.01 and mean centred on the value of the same gene in a model in the previous generation, g_i, subject to constraining the value to between 0 and 1. These samples were taken in genotype rather than phenotype space since this reduced the size of the space to be explored.

Evaluation started with transforming from genotypes to phenotypes. This used the genotype-phenotype predicate functions that came with the biological model, combined with the convention of translating boolean 'and' to 'min' and boolean 'or' to 'max', in order to generalise this function to continuous values in the range $[0, 1]$. The resulting value was then transformed by x^4 in order to place more of the search space over lower flux values. The result of this was used to transform the model bounds. As discussed previously, it was also important to induce some environmental variation, in order to support a diversity of solutions. This was achieved by random uniform sampling of each of a small set of selected exchange reactions, chosen along the lines of the procedure described in [157].

The selection procedure was also designed to encourage diversity of solutions. The solutions were ranked by their biomass output and by their mean gene activation, and then the bottom and top quartiles were discarded respectively. Zero biomass output (not biologically viable) solutions were also discarded even if they represented $> 1/4$ of the results. The remainder were randomly replicated to bring the total number of solutions back up to the maximum for the generation, and the next generation was constructed by mutation as described previously.

This procedure was followed for 4870 generations with a batch size of 100, for a total of 4 870 000 samples.

This flux dataset used the *iJO1366* model of *E. coli*, compared to the significantly smaller core metabolism model used in the other case studies. This made it substantially more computationally demanding to both create and to analyse in Chapter 6, but it contains features that are more realistic, in particular a larger number of reactions that are not involved in biomass production, and larger number of fully dependent reactions—pairs which share a metabolite exclusively.

3.5 Conclusion

This chapter has described the potential modification strategies that can be used in biased Monte-Carlo sampling over metabolic flow networks, and how those strategies can be conditioned in a way that is appropriate to use with metabolic network simulations. The core purpose of this chapter was to both describe and define these modification strategies, and to show how concrete sampling strategies can be derived from them which can create datasets which can yield information about metabolic network properties. As can be seen, these datasets were created, and owing to the new software described in section 3.4.1, this was possible with large sample sizes in a reasonable time frame: tens of hours rather than weeks. These dataset obviously also needed to be useful for characterising network properties, and this is demonstrated in the following chapters.

These modification strategies and conditioning themselves are for the most part fairly well known. However, the concept introduced here of biased sampling over metabolic flow networks appears to be novel, and demonstrates a new pattern that allows for analysis of metabolic networks to combine general network structure with condition specific information. Finally section 3.4 described case studies covering a variety of these techniques, which are used both for validation and as examples in Chapter 5 and Chapter 6. The datasets created in case studies 1 and 2 represent to my knowledge the largest datasets of their kind in existence, at approximately 500 000 samples each.

CHAPTER 4

Small sample approaches

4.1 Introduction

This chapter describes approaches that I have developed to aid in the understanding of small sets of metabolic networks—techniques to interpret structure of single networks, to compare pairs of similar networks, and to work with small single digit collections of networks derived from small gene expression data sets.

This forms something of a counterpoint to the approaches described in Chapter 5 and Chapter 6, since with small sample sizes the challenges are somewhat different. With small sample sizes, finding statistically significant features of the datasets is much more difficult, but with the correct tools it is possible to have a greater degree of human involvement. The sections in this chapter are organised moving from small collections of networks (section 4.2) to analysing single networks (section 4.4) because as the sample size gets smaller, more analytic and visual options become available.

One hypothesis of this chapter is that there is useful information to be inferred via various techniques for metabolic network comparison, even with small numbers of models compared to the other chapters. The various projects differ in their approaches to doing this, with a variety of more specific aims. An important thread of discussion of visualisation techniques for metabolic networks, particularly for differences between them. Although I created some different approaches to this, my overall conclusion was that visualisation on its own is not enough, and that more sophisticated statistical methods, as discussed in the following chapters, are required. This is a human limit as much as a software limit, since regardless of how effectively the information is displayed, systems with even tens of components are hard to understand when they have high large numbers of interactions between them, let alone systems of thousands.

The overall result of this chapter it to show that the techniques in the following chapters are needed: techniques which require detailed human comprehension do not

scale well for large networks.

4.2 Comparison across 11 samples from 4 gene expression conditions

In this project I worked on data regarding Salmonella infection in mice, specifically regarding differences between how Salmonella defends against the immune system. The sequence count data set consisted of 11 measurements for each of 6579 genes: three technical replicates for each of the three *in vivo* conditions, and two technical replicates for the *in vitro* input condition. The dataset structure is illustrated in table 4.1.

The aim here was to develop an approach that could construct an consistent explanation for the similarities and differences observed between the replicates in terms of the metabolism that was consistent with the measured gene expression. This was accomplished by fitting metabolic networks to the gene expression data available from each of the replicates. These metabolic models needed to provide a strong prior in order to compensate for the low numbers of replicates, and the overall data pipeline needed to be adjustable in such a way as to find only the most strongly supported insights, without having too many parameters and risking overfitting the model to already identified explanations.

↓ Gene locus	Group 1			Group 2			Group 3			Control	
Replicate →	1	2	3	1	2	3	1	2	3	1	2
FQ312003.1012	3	4	2	6	1	0	0	6	7	61	105
FQ312003.1187	8	9	0	8	4	4	0	0	10	1025	793
FQ312003.1229	0	0	2	1	0	0	0	4	5	119	128
FQ312003.1320	2	0	0	0	0	0	0	0	2	75	41
FQ312003.1463	6	4	0	0	0	2	2	0	0	259	165
FQ312003.1509	0	3	0	0	0	0	2	0	4	31	47

Table 4.1: A sample of the start of the Salmonella gene expression dataset. This illustrates the structure: three experimental groups, with three replicates each, plus a control group, with two replicates. Groups are listed in the first row, whilst replicates are listed in the second row. There is significant structural variation in the expression counts between groups and replicates, necessitating aggressive normalisation.

In the terminology introduced in Chapter 3, this means that this was a case of modifying reaction rates directly, but with a carefully designed expression-flux function and feedback system, in order to be able to find the smallest gene expression based modification that would produce noticeable results in the metabolic network. The first steps towards this were several layers of normalisation applied to the gene expression data.

4.2.1 Preprocessing and methods

Gene expression data consists of count data that shows large amounts of variation between technical replicates, owing to variation in the sample preparation, but within technical replicates it shows less variation and a large amount of positive skew. It was therefore important to create a normalisation approach for this gene expression data which would remove as much technical variation as possible, while maintaining the variance required to create a metabolic model which uses large parts of its network, as described in section 3.4.

After experimentation with a large variety of different normalisation approaches, a log-normal distribution was assumed for each technical replicate. The first step was therefore to take the natural logarithm of the each count plus one (since there were some zeros, particularly in the *in vitro* experiments where the values are generally much smaller). Each technical replicate was then divided by its overall mean to compensate for differences in sample preparation and materials, and then the mean count was found for each gene in each experimental group, across the technical replicates. Normalisation was conducted for each gene against the *in vitro* input as a control, and the standard deviation was adjusted to 0.1, to achieve an overall distribution which was approximately normal with mean 1, apart from a spike at 1. Figure 4.1 shows the results of this normalisation.

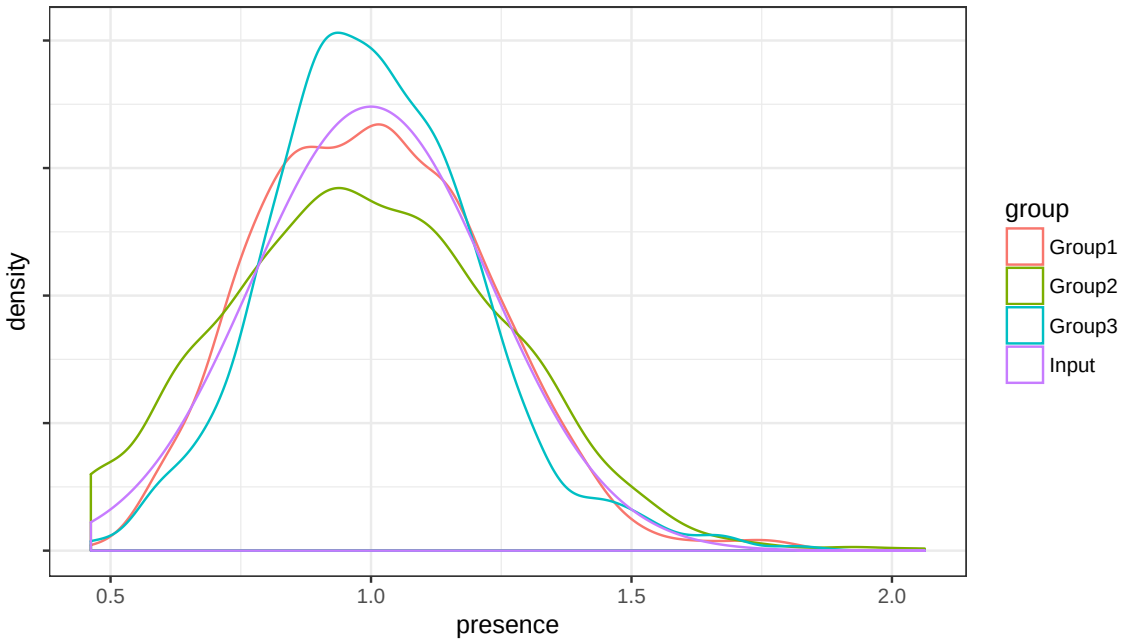


Figure 4.1: Kernel density estimate showing distribution of gene expression values after normalisation. Groups 1, 2 and 3 (red, green and blue respectively) are *in vivo* experimental groups, whilst Input (purple) is the *in vitro* control.

A standard metabolic model from the literature was used [158], which featured Gene-Protein-Reaction mappings which could be used to specify which genes affected which reactions. To combine this with the RNA data, each gene in the RNA dataset was

mapped to the equivalent gene in the metabolic model. Where multiple genes correspond to a particular reaction, they were combined via a continuous extrapolation of the boolean function encoded in the Gene-Protein-Reaction mappings. Specifically, if *both* of the genes or gene sets are required (i.e. an AND relation), the result was the minimum activation of the two; if *one* of the genes or gene sets are required (i.e. an 'OR' relation), the result was the maximum activation of the two. Where there was missing information about a gene, the gene was assumed by default to be abundant, leaving the other gene in the expression to dictate the output level. The resulting values were termed 'reaction activations'.

The reaction activations resulting from combining the gene expressions were still single values, approximately normally distributed around 1. By comparison, reactions require upper and lower constraints to control their rate, and appropriate settings for these values vary widely. The functions to map from reaction activations to constraints therefore should be defined conditionally on typical rates for the reactions of interest, but given a certain typical rate, the functions should be monotonic in reaction activation.

The approach taken to define these activation to constraint functions was to first conduct flux balance analysis using the very liberal constraints that existed in the model by default, in order to establish a 'base' rate for each reaction. This 'base' rate was then multiplied by the reaction activation to create a 'target' rate, and the constraints were set to α each side of this target. $\alpha = 10\%$ was found to be a reasonable value for this constraint width via a manual parameter search. Some exploration was conducted into using a gradient descent approach to find an optimal value for α, by finding a value that was as small as possible whilst still producing a viable model. However, this was found to produce little increase in the result quality over a hard coded value, whilst increasing the complexity of the whole system considerably, and making the tuning of other parameters slow and unpredictable. Finally, based on these new constraints, a second round of flux balance analysis was run to find the final flux values, taking 50 repeats with reorderings in order to detect under determined values.

4.2.2 Postprocessing and results

The results of the previous section were four flux distributions: three experimental, and one control. This represents over 4000 different flux values, so significant further effort was required to find the most interesting parts of this dataset. In this project a heuristic approach was taken to this problem. First, reactions were removed that fit one of a set of deselection criteria:

- all reactions where flux values were identical across experimental and control groups,
- all reactions that were under determined by the model (standard deviation $>1 \times 10^{-6}$),
- all reactions where the base flux was under determined (index of dispersion $>1 \times 10^{-3}$),

- all reactions in the exchange subsystem.

Then the top 30 reactions were selected, ordered by the standard deviation of the difference between the experimental fluxes and the control, divided by the reaction constrain width.

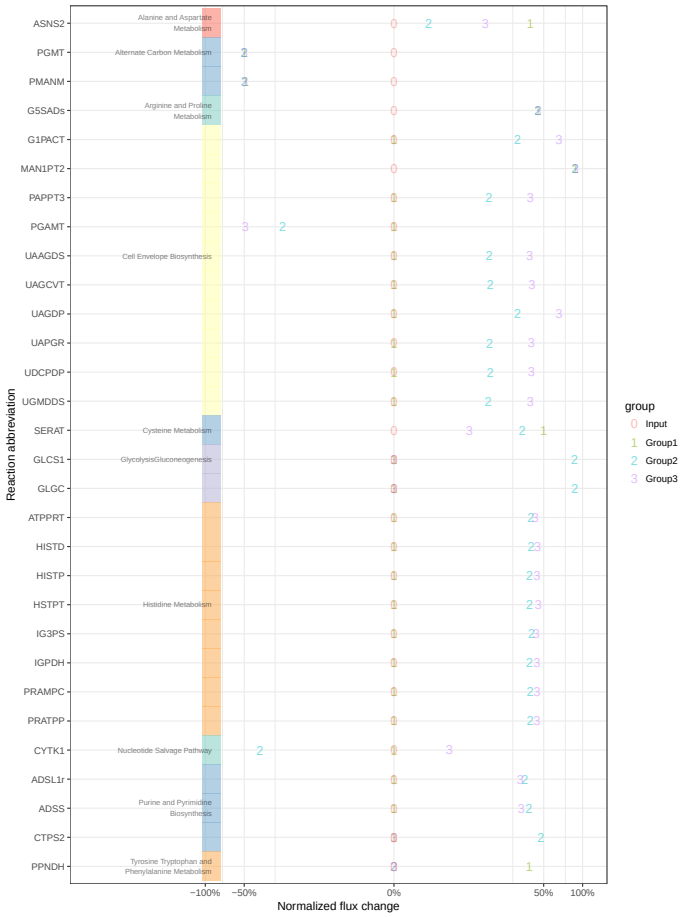


Figure 4.2: Plot of normalised flux change for a number of highly regulated reactions. Numbers show experimental groups, y axis shows reaction ID, and x axis shows normalised flux change.

After selection these reactions were plotted first using a dot plot (see figure 4.2), and then using an adjacency matrix based heat map plot (see figure 4.3 and figure 4.4).

Figure 4.2 shows on the y-axis the top 30 highly regulated reactions, grouped by their subsystems (the coloured, labelled bar). The x-axis shows the flux change for each group (shown by coloured numbers) against the *in vitro* control, which is labelled at as 0 and fixed at zero due to this normalisation.

The heat maps, figure 4.3 and figure 4.4, consist of part of the adjacency matrix of the metabolic model (reactions against metabolites), overlaid with flux differences between groups and against control, respectively. This is much easier to understand by eye than a list of reaction equations, whilst remaining readable with a larger number of elements than would be possible for a network diagram.

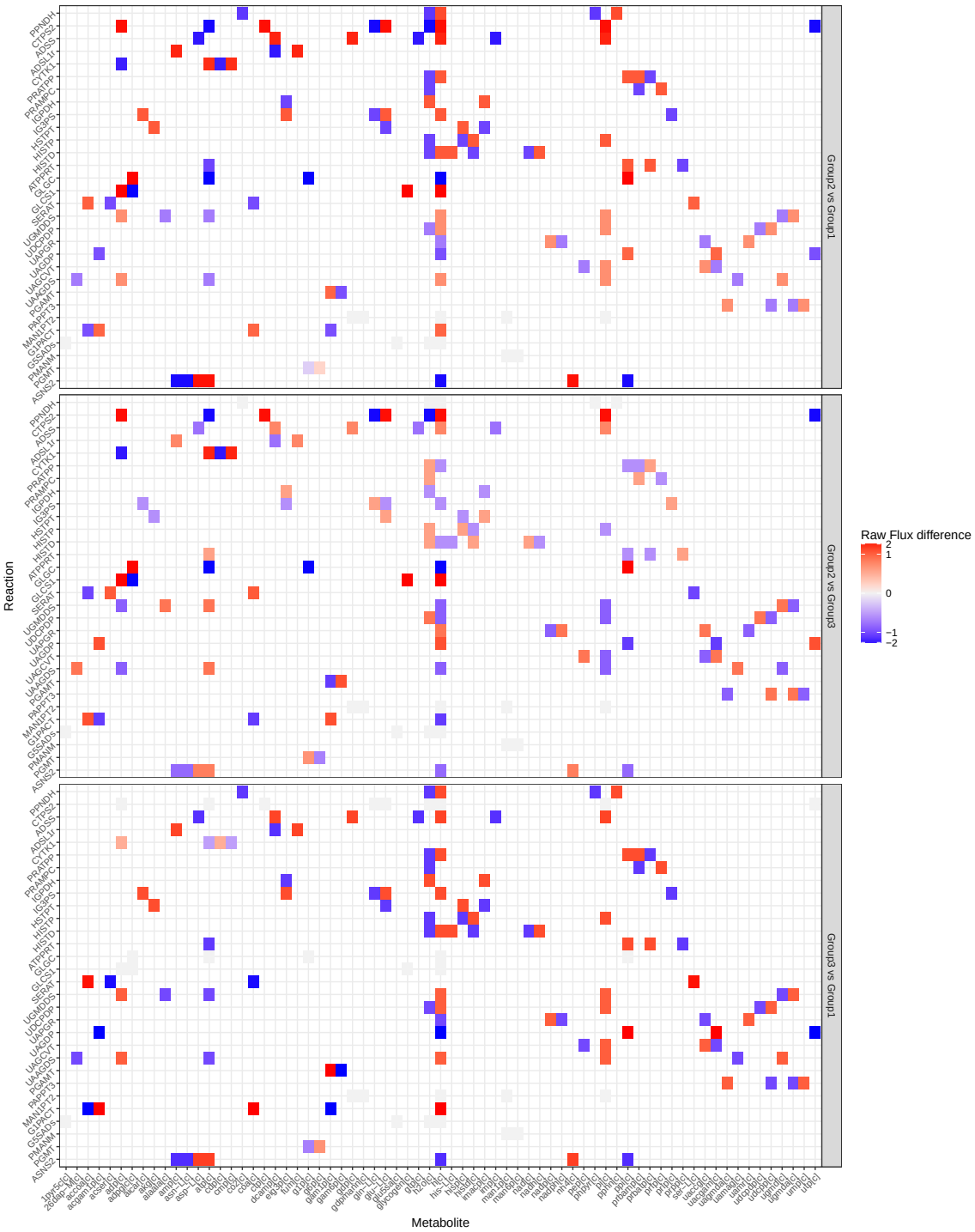


Figure 4.3: Heat map showing raw flux change, compared by experimental group. Colour shows change in flux, x axis shows metabolites, y axis shows reactions.

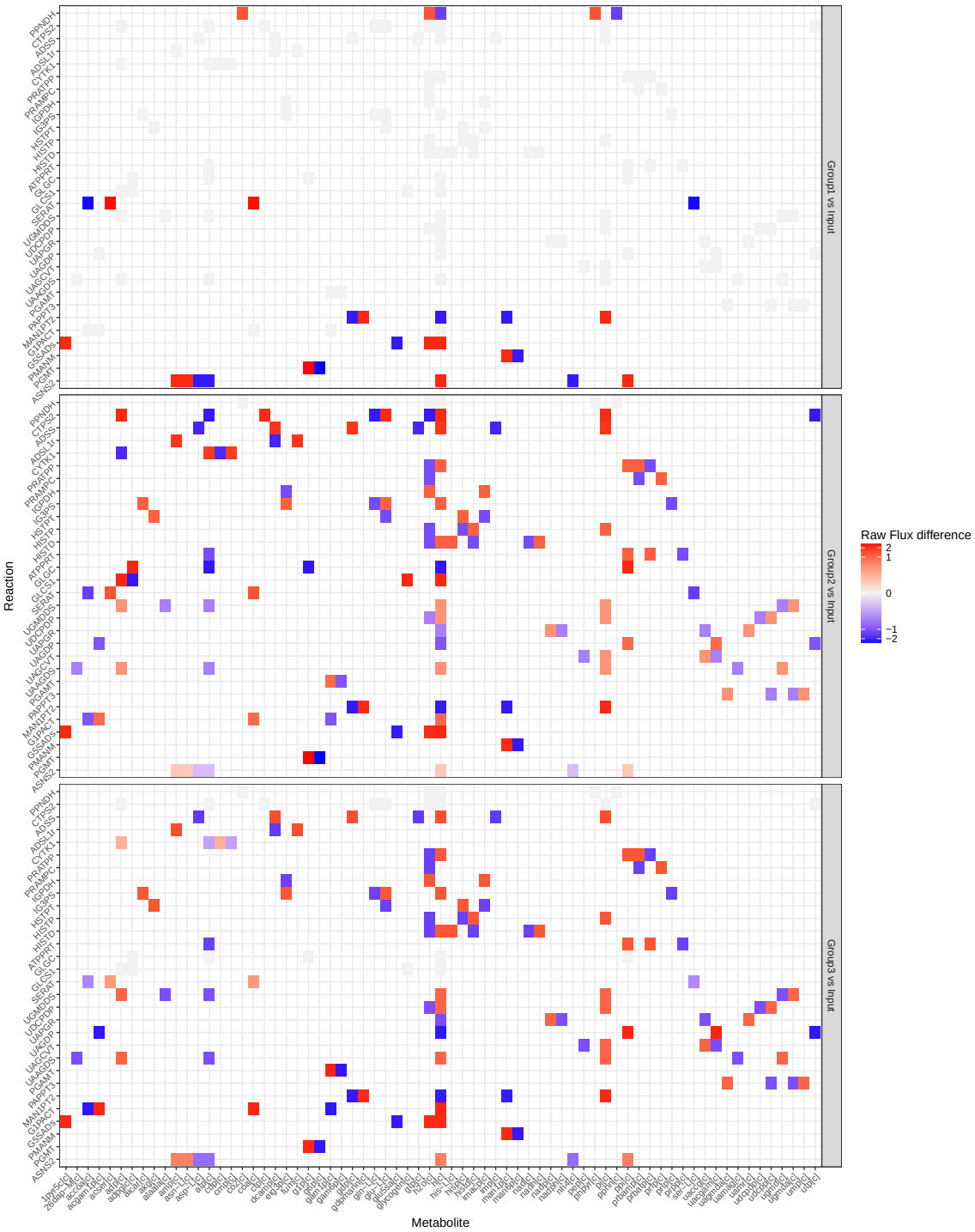


Figure 4.4: Heat map showing raw flux change, experimental groups *vs* control. Colour shows change in flux, x axis shows metabolites, y axis shows reactions.

4.2.3 Conclusion

The results of this study supported existing theory and data around the effects of the biological interventions conducted, particularly by demonstrating the growth pathway up-regulations which lead to the increased growth rate seen in bacteria hosted by gp91(phox) deficient mice, such as the increase in the use of Phospho-N-acetylmuramoyl-pentapeptide-transferase, an enzyme which participates in cell wall synthesis. [9]. The results of this study were published in *Transcriptome and proteome analysis of Salmonella enterica serovar Typhimurium systemic infection of wild type and immune-deficient mice* [9].

From a technical point of view, figures 4.3 and 4.4 represent, to my knowledge, a new way of plotting changes in flux between related metabolic networks. They highlight reactions and metabolites with substantial differences effectively, and are capable of still being readable when displaying relatively large portions of the network. However, they are weaker in terms of wasted space, since a typical metabolic network is relatively sparsely connected.

Overall, this work succeeded in the aim of producing consistent explanations for the observed biological effects, and moreover proved the hypothesis that this was possible, and a viable method of knowledge generation, since the predictions of the fitted metabolic model mirrored separately identified biological results.

4.3 Pairwise statistical and visual comparison via evolutionary optimisation: Metabex

In *Iterative Multi Level Calibration of Metabolic Networks* [10], I worked on the comparison of two closely related species in the Proteobacteria genus Geobacter, which have potential applications in electricity production. The datasets explored here were the results of an evolutionary multi-objective optimisation procedure. For each species this optimisation procedure produced a Pareto front—a large collection of models with varying gene expression values, representing points along the tradeoff between two metabolic objectives, in this case biomass production and a proxy for electricity production. A simple step plot (see figure 4.5 for an example) of the two objectives against each other can be used to visualise the general shape of the Pareto fronts, but understanding the underlying features that cause the shape is much more difficult. However, the intended application of this research was the creation of an automated system to suggest avenues of exploration for metabolic engineering, and so it was very important to identify smaller sets of useful modifications, since these are easier to test. The most important hypothesis here was not about the multi-objective optimisation component itself, but the information available in the resulting dataset of Pareto optimal models. Specifically, hypothesis was

that the information available in Pareto front datasets could be used to generate meaningful insights into cell behaviour. There was also the hope that this could be used in a semi-automated fashion to allow exploration of metabolic networks with some guidance about important features.

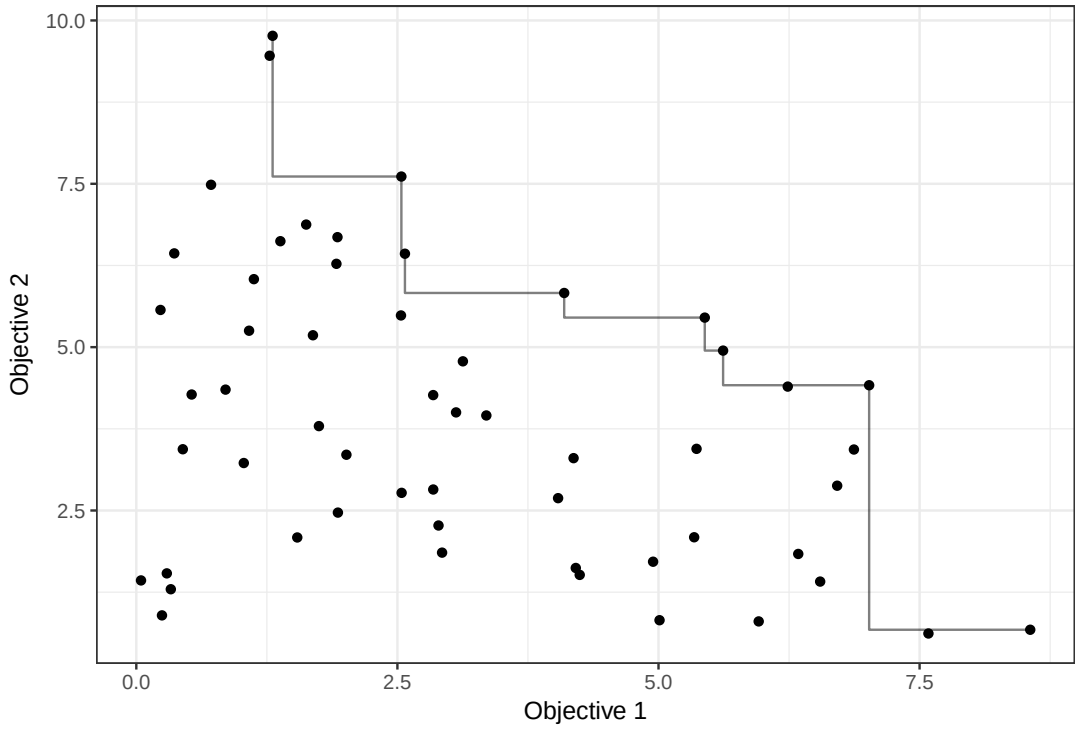


Figure 4.5: Sketch of step plot of Pareto front. The points represent potential objective tradeoffs, whilst the line connects the subset of these points that are not Pareto dominated by any other point—the Pareto front.

In this project I tackled the goal of explaining the variation in the network over the datasets from a correlation based standpoint. Starting with raw simulated expression data, such as visualised in figure 4.6, I first fitted a line to the tradeoff between metabolic objectives represented by the existing Pareto front, and then measured the correlation of each reaction activation to this line. After aggregating by subsystem labels, this generated plots such as figure 4.7.

This provides a signal for where to look for the most important reactions, and a more time consuming manual review of each interesting looking subsystem can yield a smaller set of reactions of particular interest, but understanding how these reactions interact together is more difficult.

Network plots are the clearest way of presenting small metabolic models or parts of models, but by contrast, for larger models these plots are completely unreadable, as more edges necessarily cross each other, and any pattern in the positioning of nodes becomes far more dependant on the whim of the layout algorithm used than on the actual structure of the data. Therefore, it is very important to view interesting reactions in the context of

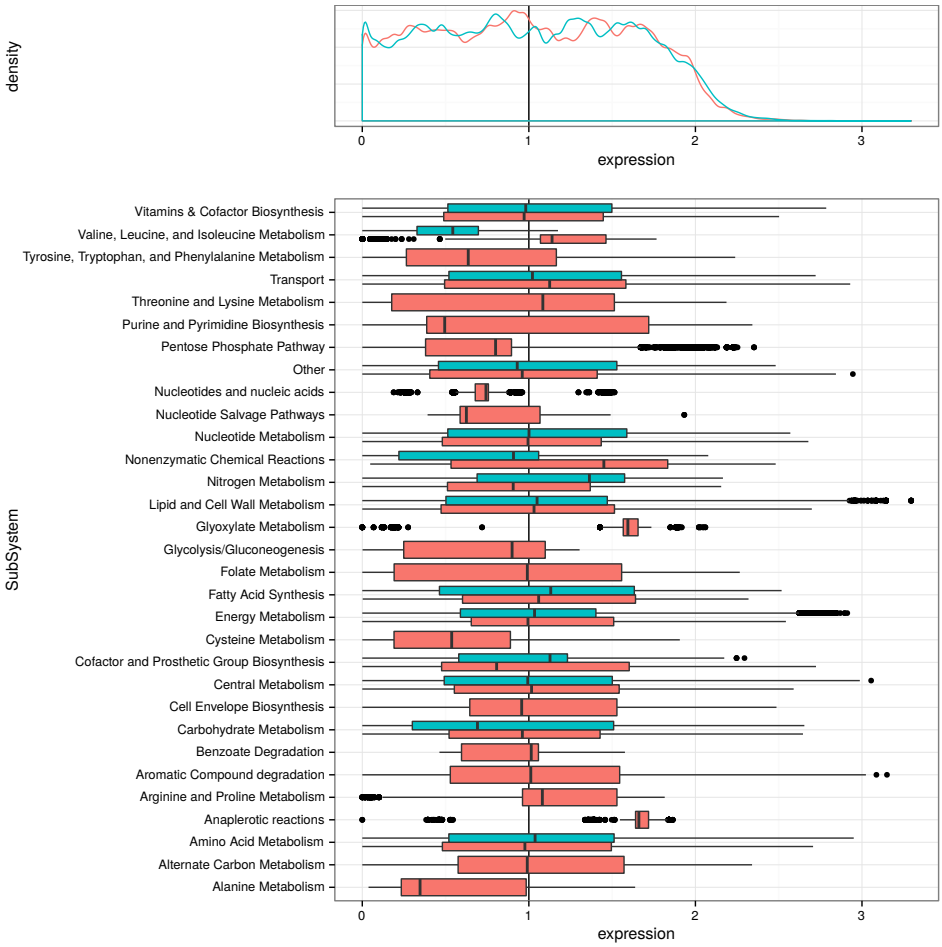
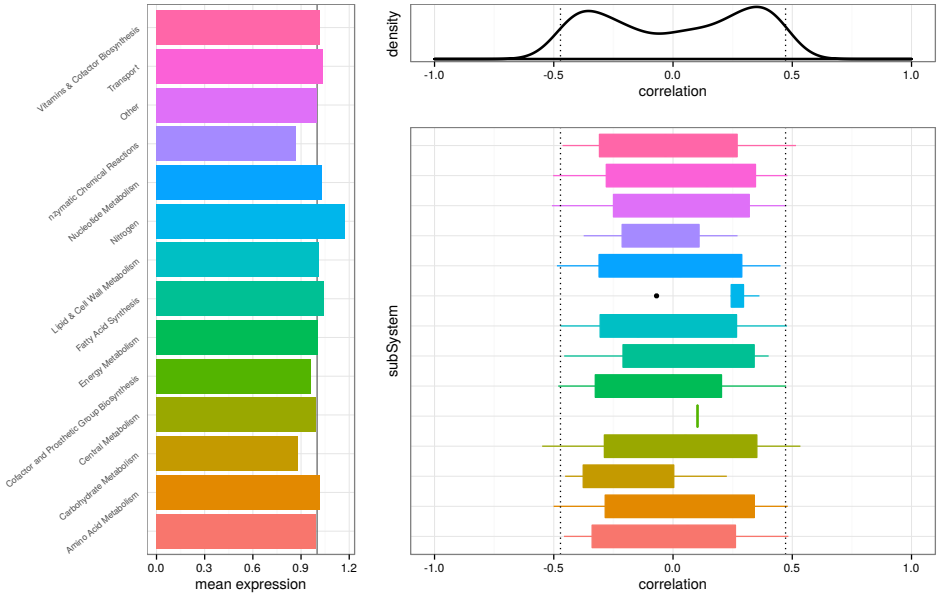
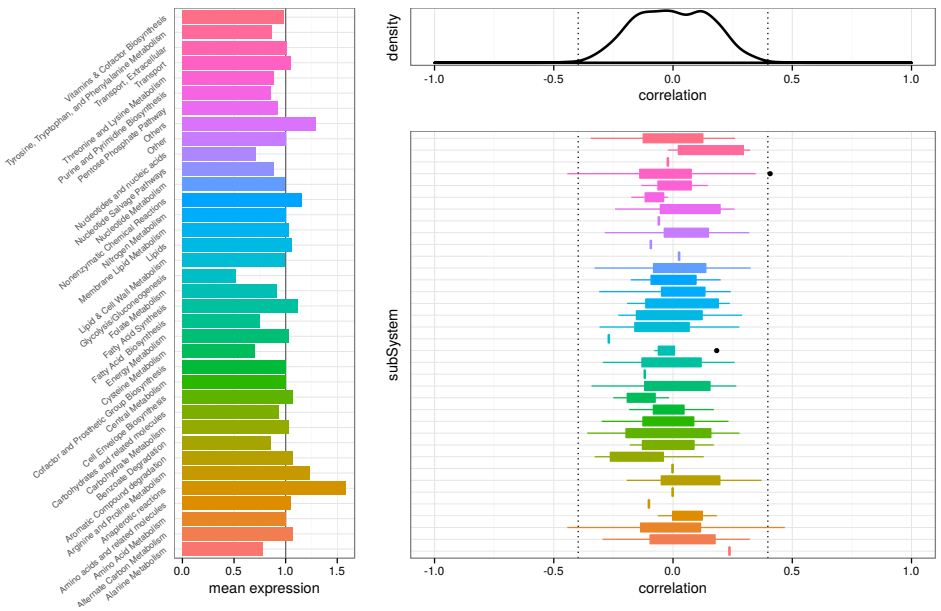


Figure 4.6: Plot showing expression change across subsystems in two *Geobacter* species. The colours show species; *G. sulfurreducens* is blue and *G. metallireducens* is red. The density plot at the top shows the distribution of expression levels. Expression levels expressed in fold change from wild-type, so zero means no expression, one means unchanged level from wild-type, and 2 means expression at double the wild-type level. The box and whisker plot shows the genes, aggregated by subsystem, on the same x-axes as the density plot. All subsystems are up or down regulated significantly ($p < 0.01$). We can see that where subsystems are labelled in both models, they are typically regulated similarly. Where larger differences exist, such as in Valine, Leucine and Isoleucine metabolism, this is due to the subsystems being quite small, so that small differences in the reaction sets included can create relatively large differences in overall regulation. This figure differs from figure 4.7 in that this figure shows raw expression, whereas figure 4.7 shows correlation between gene activity and phenotype.



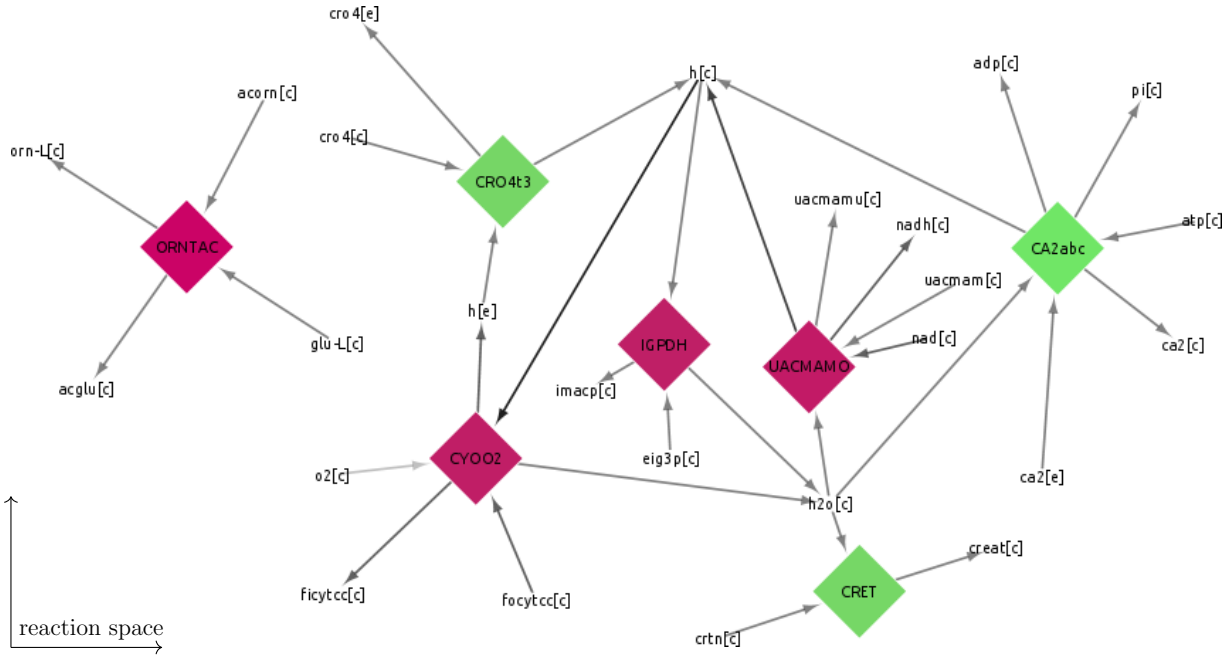
(a) *Geobacter sulfurreducens.*



(b) *Geobacter metallireducens.*

Figure 4.7: The density plot at the top of each figure shows the distribution of correlation between expression level and position in the Pareto front. The Gaussian component to the correlation distributions shows the effect of genetic drift, whilst in figure 4.7a, we see marked selection away from 0—this indicates that most of the genes have either a positive or a negative effect on the phenotype, and so most face selection pressure in one direction or the other. The box and whisker plots show the genes, aggregated by subsystem, on the same x-axes as the density plot. The bar charts to the left act as a key, and show the mean expression levels for each subsystem. This figure differs from figure 4.6 in that this figure shows correlation between gene activity and phenotype, whereas figure 4.6 shows raw expression.

only their local neighbourhood, and to allow a great deal of manual input into selecting both what is shown, and how it is laid out. In this case I decided to base a model viewer on an existing network visualisation package, Cytoscape, in order to gain access to fast rendering and interaction features, whilst using Cytoscape's application programming interface in order to statistically filter and preprocess the data to be viewed. I named this postprocessing and visualisation platform *Metabex*. This enabled the creation of plots such as figure 4.8, where a subset of reactions to view was selected by the correlation based approach described earlier, transformed to a bipartite graph representation and pushed to Cytoscape via the application programming interface, with annotations to specify a visual presentation to make the underlying data clear.



Abbreviation	Name
CA2abc	calcium transport via ABC system
CRET	creatininase
CRO4t3	chromate transport out via proton antiport
CYOO2	cytochrome-c oxidase (2 protons translocated)
IGPDH	imidazoleglycerol-phosphate dehydratase
ORNTAC	ornithine transacetylase
UACMAMO	UDP-N-acetyl-D-mannosamine oxidoreductase

Figure 4.8: Network visualisation of a subset of reactions from *G. metallireducens* and associated key. This subset is derived via a variant on the procedure shown in Figure 4.7. Nodes in burgundy indicate positive correlation with Fe^{2+} synthesis, whilst nodes in green indicate negative correlation with Fe^{2+} synthesis. Arrow weight shows stoichiometry. The stoichiometric network shown here only represents a portion of the metabolome. Underlying this metabolic network is the genetic network of genes and their links to the reactions that they regulate, in a multiplex manner [159].

4.3.1 Conclusion

This project fulfilled the goal of providing a comparison between metabolic networks, but something was left to be desired in terms of usability and technical maintainability of the Metabex tool itself, as a result of a great deal of interactions between different open and closed source packages. However, this did confirm to me that adding a degree of user interaction to network plots made them substantially easier to understand, since it is possible to look at different parts of the network without worrying so much about reactions overlapping each other. The first hypothesis, that it is possible to use Pareto fronts to identify meaningful behavioural information appears to be true. This is particularly evident from figure 4.8, where it was clearly identified that chromate and calcium exchange are highly correlated with Fe^{2+} synthesis. This is not a particularly surprising result in itself, but the method by which it was obtained is novel and showed great potential, since this effect is not directly visible from the metabolic network structure. The secondary goal, to base a semi-automated tool upon this technique, was less successful. Although the statistical component was effective, the actual user interaction achieved in this attempt left much to be desired, and was greatly improved upon in section 4.4.

4.4 A platform for examining small numbers of models: FBAonline

In this project (*MetCiliates: a ciliate meta-model multi compartment resource for the analysis of the free-living ciliates metabolism. A Tetrahymena thermophila case study.* [11]), I worked in a collaboration to create and analyse a new model of the free living ciliate *Tetrahymena thermophila*. This revealed that whilst these kinds of large scale metabolic models are becoming widely available and accessible in online repositories such as BiGG (`bigg.ucsd.edu`), access to even relatively simple analytic approaches is much more restricted. For instance, the current most popular toolbox, Cobra [160], requires the installation of Matlab and Git, the manual installation of a linear programming solver, followed by obtaining the toolbox itself using the Git 'clone' command and a Matlab install script. For a user familiar with the relevant packages, this installation procedure is involved and error prone, whilst for someone who is primarily interested in investigating only one particular organism, this represents an unreasonably high barrier to use.

My Fbar (see section 3.4.1) package is designed to deal with some of these difficulties, and after the installation of R, Fbar itself is installed with one line. However, overall Fbar is intended to be targeted specifically towards more technically able users, as it is designed with a small API to make it easy to include as part of large analytical pipelines.

This meant that for the Tetrahymena project, the aim was to create a platform that

would have a lower barrier to entry than Fbar, and to provide some of the analytical techniques I had developed with Metabex, but without the associated usability and maintainability difficulties.

My solution to this was to create the web application *FBAonline*. This uses Fbar as a back end to provide a browser based tool for analysing metabolic models. In addition to basic flux balance analysis, it provides:

- flux rate comparison similar to that used described in section 4.2;
- network visualisation similar to that provided by Metabex's Cytoscape integration, using the Vis.js JavaScript library; and
- a new reaction clustering approach that works by trimming a hierarchical clustering of the full network.

Input is accepted from Google sheets—the aim here is to encourage easy collaboration on model curation. The development of a browser based tool also allows the use of relatively fast open source linear programming solvers without the potentially complex installation procedures for packages like GLPK.

The reaction clustering approach embedded in FBAonline works by using the Walktrap community detection algorithm [161] to find hierarchical community structure in the reaction connectivity network. This establishes a dendrogram covering the entire metabolic network, but this tree has several hundred leaves, and so is too verbose to read. This is therefore followed by slicing the tree into subtrees at a set height in order to select the strongest connections, and then selecting the largest of these well connected subtrees. FBAonline's interactivity is particularly helpful for choosing the best cut height, since this differs between models.

Figure 4.9 shows the results of this reaction clustering approach applied to *Tetrahymena thermophila*. The detail level (specified by cut height) can be varied by the website controls. At the level chosen in figure 4.9 we can see the identification of four sections of the metabolic model. These sections focus on phosphotransferase, fatty acid elongation and galactose metabolism separately, along with a section covering the overlap of hydrolyses in fatty acid elongation and galactose metabolism. These areas of the metabolism are strongly represented in the figure because they contain large numbers of similar molecules, linked by several inter-conversion reactions. At lower detail levels, these areas are linked together and other, more sparsely connected areas are shown in addition.

The techniques embedded in FBAonline covered the requirements for the project it was created for, and cover the majority of requirements for the typical metabolic network analysis projects that I have experienced. These analyses are restricted to those that are fast enough to be interactive, since the amount of time people are prepared to wait for

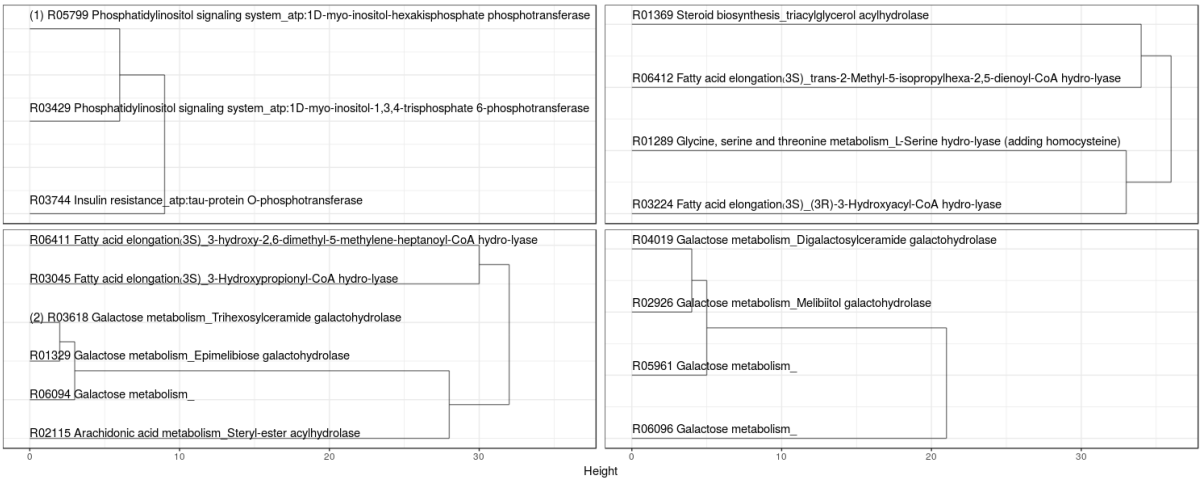


Figure 4.9: Clusters of reactions in Tetrahymena thermophila metabolic model found by tree cutting approach.

a web application tends to be less than even a graphical local application. This means that the techniques described in the later chapters are too complex to be embedded in FBAonline directly. However, the possibility exists of precomputing datasets for widely available published models, which could allow them to be explored via this tool; this is discussed further in section 7.2.

4.4.1 Conclusion

This project was successful in that it was indeed able to demonstrate that comparative and explorative flux balance analysis tools could be provided in a much easier to use manner, including the ability to make comparisons between models and take advantage of the resultant information. One limiting factor on this success was of course the requirement of easy, real time usability itself, because this limited the tool to only comparing model pairs, rather than being able to use statistical techniques with multiple models. In terms of the visualisation component of the tool, the limiting factor was found to be human as much as technical. The network visualisations were good enough for viewing elements of the network that were already known to be significant, but the overall system is too complicated to understand in this piecemeal manner.

4.5 Conclusion

This chapter has described three studies in inferring latent structure in metabolic networks across relatively small numbers of models, showing that at least some information about biological network structure can be inferred even without the larger datasets used in the other chapters. Section 4.2 demonstrated that metabolic network comparison can be

used to generate biologically relevant insights, by showing how it can directly describe the mechanism for increased growth under particular conditions. In section 4.3, I showed how the analysis of both individual metabolic networks and pairs of networks can be enhanced by the use of Pareto optimisation to generate large sets of examples against which to measure the correlation of various properties. In this section I also introduced the advantages of closely coupling statistical and graphical network analysis, in order to make it easier to understand the results of complex network analyses, and in section 4.4, I demonstrated how such techniques can be made more approachable and user friendly, such that they could be easily shared with others.

This represented the first use of any kind of metabolic network analysis on the datasets described in section 4.2 and section 4.4 and the first software specifically for metabolic network comparison in section 4.3. In addition all of the software (Fbar, Metabex, and FBAonline) is novel in its design and instantiation.

In terms of the visualisation approaches discussed in this chapter, the central conclusion is that visualisation alone is not enough. It seems that there is progress to be made in the field of metabolic network visualisation, and the tools created here, particularly FBAonline, are as good as any available, but there is a tradeoff between the size of the network community and the clarity of presentation, and a human component that limits the complexity that can be understood. These weaknesses of visualisation tools form part of the motivation for the direction of the rest of this thesis, since if network structures cannot be directly detected by visualisation, statistical tools are required.

The following chapters address approaches where much larger sets of models were used, with more heavily automated unsupervised approaches.

CHAPTER 5

Network interpretation by Similarity Network Fusion

5.1 Introduction

This chapter discusses my use of Similarity Network Fusion (SNF) to integrate data from a population of *E. coli* models representing different states of the genotype and phenotype.

Similarity Network Fusion is a network integration technique introduced in 2014 in the paper *Similarity network fusion for aggregating data types on a genomic scale* [6, 162]. It is designed to merge multiplex networks (sets of networks with the same nodes but different edges) into simple single layer networks. Naturally, this ability to merge data from multiple sources is relevant to multi-omic research, which often involves the situation of having multiple links between corresponding entities.

In the paper *Multiplex methods provide effective integration of multi-omic data in genome-scale models* [12], I applied SNF to a multi-omic dataset for the first time.

The overall hypothesis that this chapter aims to prove is that it is possible to extract information about population structure from metabolic network datasets, derived both from experimentation and random sampling using the techniques described in Chapter 3, and furthermore to show that this population structure is at least somewhat derived from the network itself, rather than merely the sampling strategy. In the context of this chapter, population structure means structure, such as clustering, that exists in the comparisons between different metabolic networks in the dataset (*versus* network structure which exists in the comparisons between nodes and edges inside metabolic networks).

This population structure was identified successfully using the Weighted Similarity Network Fusion technique, followed by application of spectral clustering. Clustering techniques can have a tendency to find structure from noise, meaning that it was important to validate that this structure was real. This was first achieved by repeating the technique on

both real and simulated datasets, testing my assertion that the population structure was at least partially intrinsic to the network structure itself. Then, this overall population structure was shown to be connected to Fe^{2+} fluxes in both datasets, highlighting that since the underlying causes for the structure were related, this was extremely unlikely to have occurred by chance.

This is particularly interesting because these types of clusters would normally be attributed to genetic variation, or a bistable regulatory mechanism, rather than to being implied by the metabolic network itself.

This chapter first introduces some background and theory on SNF before discussing how I applied it to multi-level metabolic data, and how the core algorithm was modified to tailor it to this application. The resulting technique is then applied to an experimentally derived gene expression database, and to a simulated dataset from Chapter 3, resulting in the detection of similar quantised patterns in both datasets.

5.2 Overall approach

The overall aim of this piece of work was to demonstrate methodology that could be used to interpret multi-omic biological data, and understand more about the genotype-phenotype relationship.

To conduct this research a high quality starting data set was needed, which was based on two resources: the Colombos gene expression database, and the *iJO1366* model of *Escherichia coli*. The Colombos database contains whole genome gene expression values for 2369 different conditions of *E. coli* growth, whilst the *iJO1366* model is a recent metabolic model of *E. coli* which contains 2583 reactions.

From these two resources, the complete genotype-metabolome-phenotype data set was created by simulating metabolic activity using *iJO1366* for each condition in the Colombos database, using *min* and *max* functions in place of AND and OR to convert gene predicates in the model into real valued functions, and then converting the results of this into reaction bounds using a logarithmic function similar to that in equation (3.26) on page 48.

This created a resulting population of 2369 individuals, where each individual consisted of a full gene expression vector, and a simulated metabolic flux vector created from the gene expression vector. In addition, a phenotype was extracted based on the acetate production rate of each individual.

In order to analyse this data using SNF, it had to be in the form of a multiplex network. The first and most obvious network representation would be to use the metabolic network itself to provide network structure, with genes as one layer, reactions as another, and links quantified by some sort of similarity measure across all individuals in the population. However, on further inspection this approach falls apart, because the gene-reaction

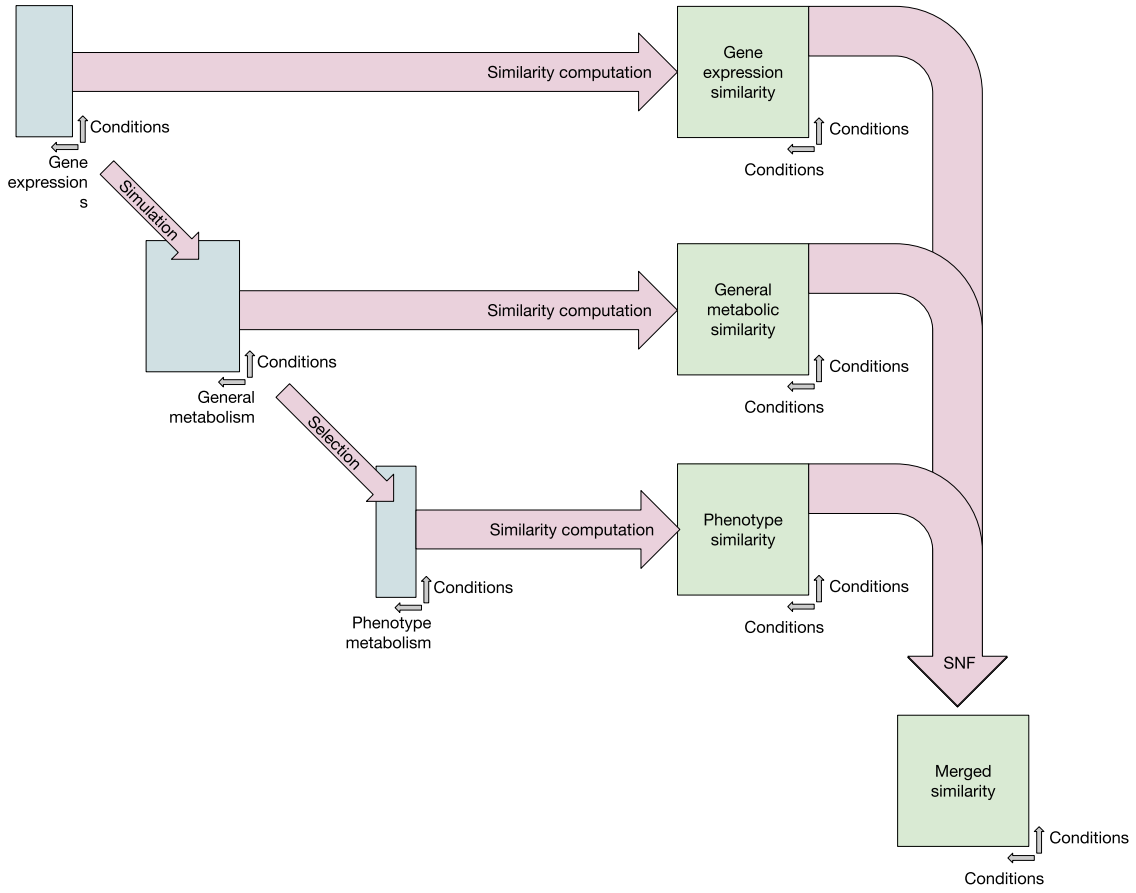


Figure 5.1: Flowchart of data types and processes. The process begins with gene expression data. From this gene expression data, metabolic flux data is generated for each condition. Phenotype data is a biologically important subset of the flux data. Similarities between conditions are computed from each of these data types separately. Finally these similarities are combined by SNF into an overall similarity matrix.

mapping is many to many. An individual reaction often either has no controlling genes or a combination of several, and individual genes control many reactions. Instead, the similarity used in this research was similarity between individuals, so that the resulting multiplex network consisted of a fully connected network of 2369 individuals, with edges representing pairwise similarity in their genotype, metabolome, and phenotype. Figure 5.1 shows an overview of the approach taken. Similarity here was calculated using simple correlation. For a more complete discussion of similarity computation, see section 6.4, and for background, see section 2.7.2.

5.3 Background on mathematics of Similarity Network Fusion

5.3.1 Motivation

In recent years, complex networks have come to increased prominence as a dataset type. Many types of information, particularly in biology, are best represented in terms of pairwise interactions (edges) between entities (nodes), which can be a particularly compact form for large systems in which only a small proportion of elements have interesting interactions.

As more datasets are gathered and stored as complex networks, it becomes increasingly common to find multiple network datasets (edge sets) that refer to the same entities (node sets). One might suppose that sometimes these network datasets are different reflections of some common underlying network, and therefore wish to integrate the known dataset to discover the structure of the underlying, unknown, dataset. This is the purpose of Similarity Network Fusion (SNF) [6].

5.3.2 Definition of W

W are the base similarity networks that need to be integrated, represented as similarity matrices.

In different applications of Similarity Network Fusion, these can represent different quantities, but in this context they represent the similarities between individual samples in the Colombos dataset. Two of these similarity networks are used, representing similarities between the genomes and similarities between the metabolomes, represented by the internal fluxes in a metabolic network simulation based on the genotype,.

Of course, this meant that it was necessary to calculate a single scalar similarity value from the vector properties of each pair of individuals. In order to do this, I used an approach that was similar to the scaled exponential similarity kernel suggested in Wang *et al.* [6], but with extra normalisation steps which I found to be necessary. First, I removed gene expressions and fluxes for which less than 10% of values were known and finite. Then, each expression and flux was divided into 20 quantiles, and each value was replaced by their quantile number, to deal with the high kurtosis. Finally, I calculated the Euclidean distances between nodes (conditions), and found similarities $\mathbf{P}_{ij}$ in each layer v by exponential negative squared Euclidean distance

$$\mathbf{P}_{ij}^{(v)} = e^{-\left[d_{ij}^{(v)}\right]^2}, \tag{5.1}$$

where $d_{ij}^{(v)}$ denotes the Euclidean distance between the two arrays representing conditions

i and j in the vth layer.

For comparison, the scaled exponential similarity kernel suggested by Wang *et al.* [6] is defined as follows:

$$\mathbf{W}(i,j) = \exp\left(-\frac{\rho(x_i, x_j)^2}{\mu\epsilon_{i,j}}\right). \tag{5.2}$$

where ρ is the Euclidean distance function, μ is a parameter typically in $[0.3, 0.8]$ and $\epsilon_{i,j}$ is a separate scaling function:

$$\epsilon_{i,j} = \frac{\text{mean}(\rho(x_i, N_i)) + \text{mean}(\rho(x_j, N_j)) + \rho(x_i, x_j)}{3} \tag{5.3}$$

where $\text{mean}(\rho(x_i, N_i))$ is the mean distance between x_i and its neighbours.

I experimented with this version but found the results to be similar with or without the scaling function, which I suspect to be due to the aggressive normalisation which I carried out beforehand. For this reason I judged that it was better to not use the scaling function, since it added another layer of complexity without an associated improvement in outcome.

5.3.3 Definition of P_0

P_0 are the starting state of the networks, which will be modified as the algorithm iterates. As such, they start as normalised versions of W. Appropriate normalisations are:

$$\mathbf{P_0} = \mathbf{D}^{-1}\mathbf{W} \tag{5.4}$$

where

$$\mathbf{D}(i,i) = \sum_j \mathbf{W}(i,j), \tag{5.5}$$

or

$$\mathbf{P_0}(i,j) = \begin{cases} \frac{\mathbf{W}(i,j)}{2\sum_{k\neq i}\mathbf{W}(i,k)} & \text{if } j \neq i \\ \frac{1}{2} & \text{if } j = i \end{cases} \tag{5.6}$$

Equation (5.6) is more robust to numerical instabilities.

5.3.4 Definition of S

$\mathbf{S}$ are local similarity matrices for the networks. These are modified similarity matrices such that only the K nearest neighbours of a node have non-zero similarity. If N_i is the

set of neighbouring nodes of x_i (including x_i), then we define local similarity by:

$$\mathbf{S}(i,j) = \begin{cases} \mathbf{W}(i,j) & \text{if } j \in N_i \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

5.3.5 Core operation

The core network integration procedure consists of iteration of the following update equation:

$$\mathbf{P}_{n+1}^{(v)} = \mathbf{S}_n^{(v)} \times \left(\frac{\sum_{k \neq v} \mathbf{P}_n^{(k)}}{m-1} \right) \times (\mathbf{S}_n^{(v)})^T, \quad v = 1, \ldots, m, \tag{5.8}$$

where v is the index of each of the m layers, k ranges over all layers except for the one under consideration, n is the iteration number, $\mathbf{P}_n^{(v)}$ is the similarity matrix, and $\mathbf{S}_n^{(v)}$ is the local similarity matrix, both referred to the vth layer at the nth iteration.

For a set of networks (layers of a multivariate network), this iteration repeatedly makes each network more similar to the others until they converge to a single integrated network.

5.4 Weighted Similarity Network Fusion tool

In order to combine the multiple gene expression and phenotype networks together, I created a modification of SNF [6]. More technical details on the modified version can be found in section 5.4.1. SNF takes a number of networks with the same set of nodes (i.e., layers of a multiplex network), and iteratively alters each network to resemble the others, until all the networks converge to an aggregate network. However, this assumes that each layer has equal importance, which is not a reasonable assumption in this case. For this reason, I developed a weighted similarity fusion approach that allows us to account for the quality of the metabolic reconstruction when linking gene expression and phenotype. This allows the use of differing weights between layers to reflect this. For instance, in the network fusion process, if the predictive capability of the genome-scale model is high, one is able assign more importance to the phenotypic data rather than to the transcriptomic data.

Since this multi-omic model and multilevel formulation provide phenotypic flux rates from gene expression profiles associated with growth conditions, one may object that the transcriptomic layer should carry no weight in the fusion process. However, the fluxomic layer is a result of predictions of the metabolic model, and therefore it should not be considered as the only indicator of the response to a given condition. In fact, in applications to genome-scale models (as in this study), the weight represents the level of

confidence in the model itself.

Let $\mathbf{P}$ be the multiplex network of similarity between environmental conditions. $\mathbf{P}$ is computed, using (equation (5.1)) on $m = 3$ omic levels, i.e., from the distance between gene expression arrays in the transcriptomic layer, from the distance between flux rate arrays in the metabolic layer, and from the distances between the phenotype fluxes selected from the metabolic layer. The central equation of standard SNF is equation (5.8), which is iterated to actually conduct the fusion process, namely a finite number of message passing steps in which the m layers co-evolve. Equation (5.8) describes the update step for each of the m status matrices $\mathbf{P}^{(v)}$, representing similarities of conditions in each layer. These matrices are initialised as a normalised form of the similarity matrices of the m networks. $\mathbf{S}^{(v)}$ are kernel matrices, giving a normalised form of similarity only to the K nearest neighbours.

Note that equation (5.8) computes an unweighted mean over $\mathbf{P}^{(k \neq v)}$. To introduce a weighted network fusion, we introduce a vector of weights, $\mathbf{b} \in \mathbb{R}^m$, which are used to alter the update step and take a larger input from some of the m layers than others. Then, we replace equation (5.8) by performing the following update step for each layer:

$$\mathbf{P}^{(v)} = \mathbf{S}^{(v)} \times \left(\frac{\sum_{k \neq v} \left(\mathbf{P}^{(k)} \times \mathbf{b}_k \right)}{(m-1) \times \sum_{k \neq v} \mathbf{b}_k} \right) \times (\mathbf{S}^{(v)})^T, \quad v = 1, \ldots, m. \tag{5.9}$$

5.4.1 Further changes to weighted SNF tool

Conducting weighted SNF required that I created an upgraded version of the original SNF library. This is online at `https://github.com/maxconway/SNFtool`.

As well as modifying the original library to allow weighted network fusion, it was also necessary to improve the performance of the algorithm, due to the size and structure of the datasets being used. My first approach to this was to parallelise the algorithm, by exploiting the independence between the message passing steps: at each iteration step t, the edge weights in each network layer are only dependent on edge weights as of the previous step, $t-1$. This means that it is possible to achieve up to an $m-$fold performance increase when fusing m networks.

However, this speedup was not found to be enough to achieve acceptable performance, so in addition a convergence detection heuristic was implemented, to replace the original library's hard coded iteration count. At each step, this heuristic works out the mean of the normalised mean absolute deviations between equivalent values in each layer, as a measure of the progress towards fusing the networks. However, sometimes the algorithm will converge (in the sense of making negligible progress with each iteration) before this measure has reached zero, so the heuristic calculates the first and second backwards finite differences, and stops iteration if both are less than 0.01.

Mathematically, this indicator function is defined as

$$x_t = \sum_{v=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{abs(m\mathbf{P}_{t,ij}^{(v)} - \sum_{v=1}^{m} \mathbf{P}_{t,ij}^{(v)})}{n^2 \sum_{v=1}^{m} \mathbf{P}_{t,ij}^{(v)}}, \tag{5.10}$$

and we stop iteration when $|x_t - x_{t-1}| < 0.01$ & $|(x_t - x_{t-1}) - (x_{t-2} - x_{t-3})| < 0.01$.

Where x_t is the progress measure, $\mathbf{P}_{t,ij}^{(v)}$ is the value of element i, j of the vth layer of the similarity matrix at the nth iteration, m is the number of layers, and n is the number of nodes in the network.

In tests this heuristic achieved a large speed improvement over the original fixed iteration number approach, whilst being conservative enough to deliver merged results of equal quality.

5.5 Application

In the transcriptomic layer, a condition is represented by a gene expression array; conversely, in the phenotypic layer, it is represented by a flux rate array. The phenotypic flux data and, to a lesser extent, Colombos expression data displayed high kurtosis, indicating that both layers display many outliers. Outliers were not removed for two main reasons: firstly, in some cases this would remove most of the data; and secondly, this represents a biologically reasonable all-or-nothing regulation, as one would see in a bistable system created by positive feedback in regulation (for instance [163]).

After constructing the multi-omic similarity network, I fused together the phenotype and gene expression layers with a K value of 500 (number of nearest neighbours) and a phenotype-transcriptome weight ratio of 2:1. I then assessed the resulting network by conducting spectral clustering and plotting a heat map. Spectral clustering was chosen primarily because it is recommended by the authors of [162]. In the heat map, the data is well represented by three clusters. Whilst moving to larger numbers of clusters could decompose the centre cluster slightly, it sacrifices most of the contrast at the cluster boundaries. Figure 5.2 shows the heat map resulting from the spectral clustering performed on the fused network, whose outcome is reported in [12].

To detect those fluxes and expressions that are most indicative of the three configurations, I employed a regression technique based on recursive partitioning [164]. This suggested a number of decision trees with only two rules splitting on flux values, each able to assign the correct cluster in 97% of cases. I validated these fluxes as genuinely important via bootstrapping: on repeated samples of 80% of the data, the same fluxes were detected as important. The fluxes I found to be closely associated with clustering are reported in table 5.1. Of these, the eye immediately jumps to biomass generation and 5-deoxyribose production as associated with growth rate. The other exchange reac-

Reaction name	Importance (rounded %)	Cluster 1 LB	Cluster 3 UB
Biomass generation	17	1.009824	1.0079
CHEBI:44800 production	17	0.001353165	0.001350586
5-deoxyribose production	17	0.0002332694	0.000232825
p-Cresol production	17	0.0002251908	0.0002247617
CHEBI:16490 production	17	2.019649e-06	2.0158e-06
Other	15		

Table 5.1: After clustering the conditions in the case study, I built decision trees to predict the cluster from the flux rate of key reactions (and genes). This table reports those reactions that are good predictors for the clustering. For instance, conditions with low and high biomass are likely to be in two different clusters according to the value of biomass produced. The reactions shown here are therefore highly correlated with cluster membership. It appears that the clusters can be thought of in terms of a general growth rate, so the boundaries for each flux pointed in the same direction. Cluster 1 is highest growth, Cluster 3 is lowest growth, and Cluster 2 is medium growth (i.e., contained between the lower boundary (LB) and upper boundary (UB) listed in the table for clusters 1 and 3). Interestingly, two reactions (CHEBI:44800 and CHEBI:16490) out of the five predicted are underground reactions, which we hypothesise as symptoms, rather than causes, of the shifts between metabolic clusters. Since they provide alternatives to primary pathways, they are likely to be used when the primary pathways become saturated.

tions may have specific relationships with growth rate configurations, or their detection may be due to a general correlation between rates of bacterial growth and excretion of byproducts. The orange and green bars in figure 5.2 show how effective these fluxes are in partitioning the data into clusters. This indicates that the cluster structure is of high importance to the core metabolism of the cell.

5.6 Similarity Network Fusion on simulated data

Previously in this chapter, SNF was applied for multi-omic data integration, combining different kinds of biological data. However, it can also be used for exploration of pure metabolic data, by integrating information from different subsystems. Figure 5.3 shows an example of this, applied to the dataset generated in section 3.4.4. Normalisation was similar between this simulated dataset and the experimental dataset (0 mean and 0 standard deviation), and equation (3.26) was used to transform both into usable flux bounds. Distances were calculated between subsystems separately, and then the resulting 31 distance matrices were re-integrated via SNF.

Figure 5.3a and Figure 5.3b show coarse and fine grained clustering respectively—clustering into just two groups results in the 'cleanest' clustering, but targeting a larger number of groups gives a very high degree of in-cluster consistency in the rate of biomass production (green). This consistency appears to be a result of the underlying data generation mechanism: because the data was generated by an evolutionary mechanism, models are grouped into distinct 'families', descended from a common ancestor. This validates

the ability of this approach to uncover latent structure in the population data, supporting the validity of its results on biological data.

It can also be seen that the clusters are strongly connected with biomass production, but less so with other metabolic processes, such as glucose uptake, as shown in blue. This is because in this application, data was integrated from every subsystem, and biomass production requires a large number of separate inputs, from almost all subsystems, giving it a high correlation to the integrated similarity.

5.7 Validation

Showing the validity of unsupervised learning techniques is often challenging, particularly when one wants to show both correctness and usefulness—the most persuasive way to show correctness is to replicate some kind of labelled data, much like in supervised learning, but in an unsupervised learning context, it is implicit that only replicating existing results is not useful.

One disadvantage of clustered heatmaps such as those in figures 5.2 and 5.3 is that they can have a tendency to be too sensitive to clustering patterns, and appear to show clustering that is not actually there. It is therefore important to also compare these clustered heatmaps to similar heatmaps where the row and column order is based on simpler dataset characteristics. In this case the row and column ordering variables were chosen based on the biomass and Fe^{2+} fluxes. The biomass flux was chosen as the most important flux and because its reaction includes a large number of terms, meaning that it is affected by many different processes and pathways. Fe^{2+} was chosen because it very clearly exhibits an interaction with biomass that has different modes.

Figure 5.4a and figure 5.4b show these unclustered heatmaps. We can very clearly see that in both heatmaps, there are two distinct regions. In one region, there is a clear line, showing that is this region, Fe^{2+} and biomass rates are well correlated. In the other region, there is no overall order, indicating that Fe^{2+} rate is decorrelated from biomass generation in this region. We can also see that while these two modes exist in both datasets, they actually occur at different points between the real and simulated datasets. This is to be expected, since these datasets have very little in common except their underlying metabolic models; their substrates are different, which means that in one, Fe^{2+} availability is only a limiting factor at low availability, while in the other, it is a limiting factor at high availability.

5.8 Conclusion

This chapter describes the application of a modified version of the Similarity Network Fusion algorithm to a large metabolic and gene expression dataset, resulting in the detection of quantised cell states. This technique is then repeated on a simulated dataset, where similar quantised cell states are detected.

This similarity demonstrates that the quantisation is not simply an artefact of the dataset used, and therefore must be induced by the network itself (or be an artefact of the clustering procedure). However, similar quantised patterns of Fe^{2+} exchange were also detected, supporting the hypothesis that the quantisation was genuinely resultant from the network structure itself, since this could not have been caused by the clustering algorithm.

This provides clear evidence that metabolic network structure can inherently imply quantised population structure, and that analysis of large metabolic network datasets can identify this structure.

This research represented the first application of a weighted Similarity Network Fusion approach, and the first application of any Similarity Network Fusion approach to a metabolic multi-level model.

This is important because not only does it imply that Similarity Network Fusion is effective at finding population structure that derives from network structure, but that such structure exists. Typically such structure would be attributed to genetic or regulatory effects, but this work shows that it can be caused by network structure itself.

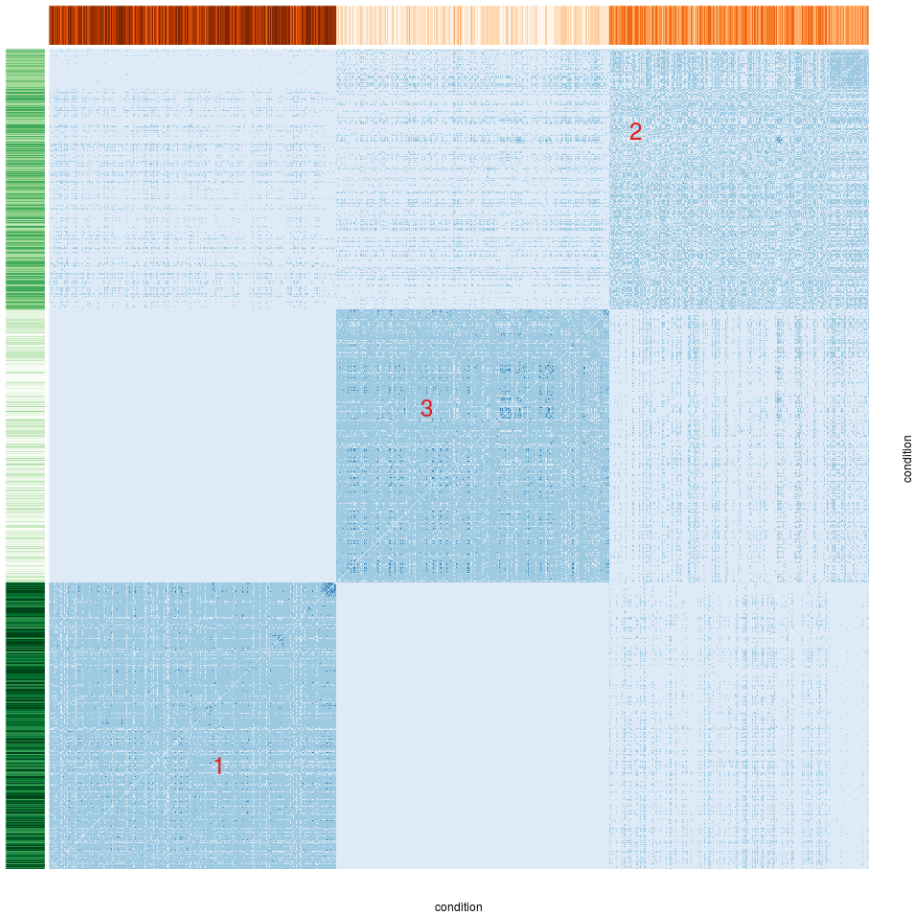
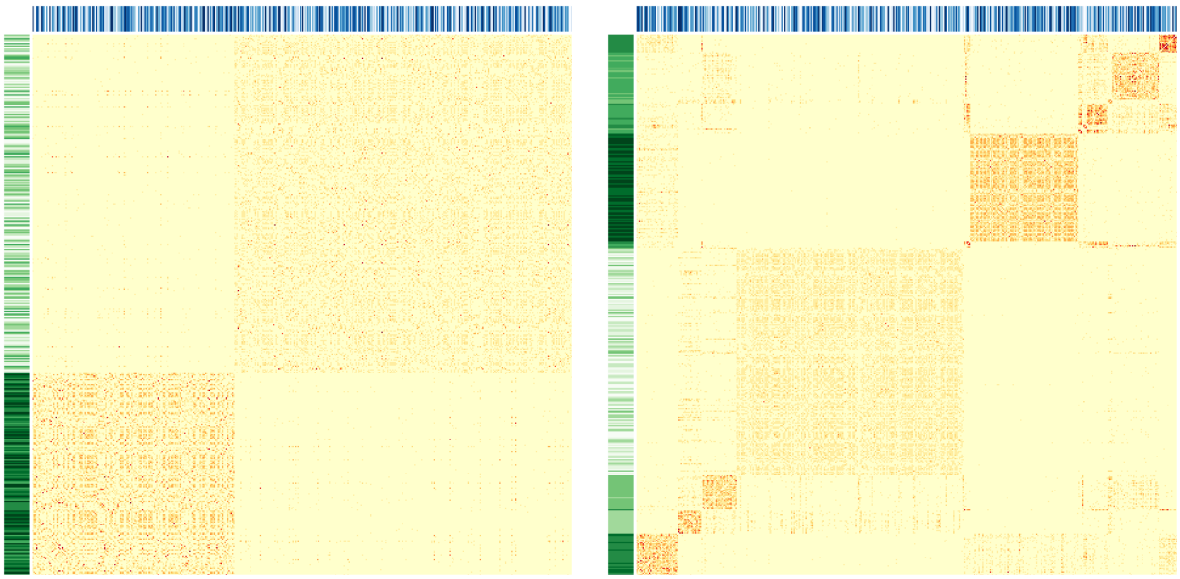
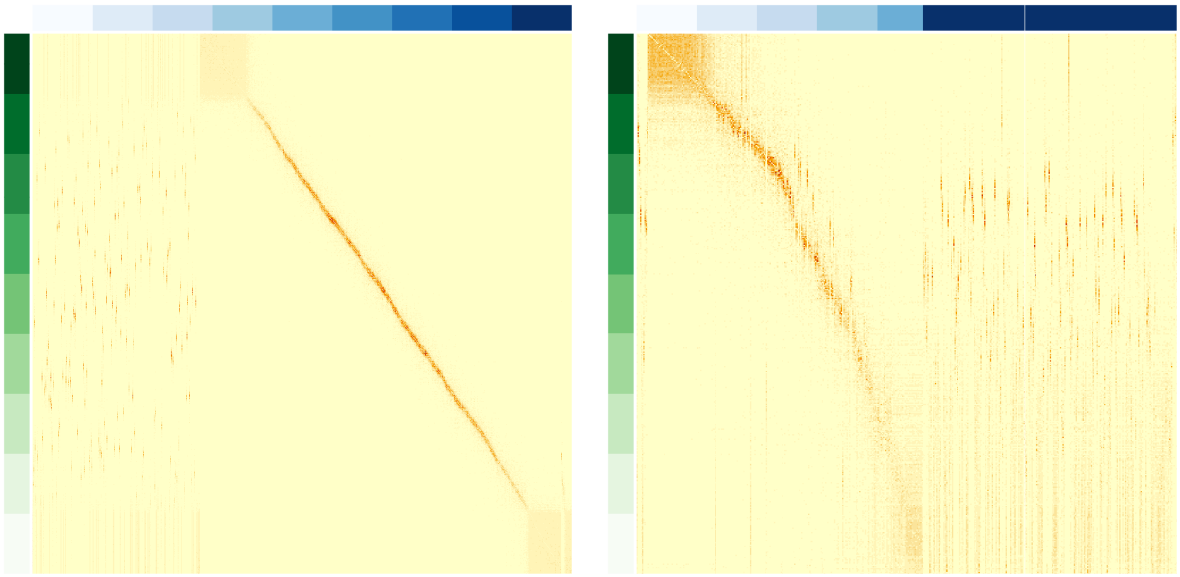


Figure 5.2: Heat map of the similarity matrix of the fused *E. coli* multi-omic network, arranged by spectral clustering into three components. The x and y axes represent the 2369 conditions, whilst the intensity of the colours in the centre represent the similarity between each of the pairs of x and y conditions. The red numbers are cluster labels, from 1 (highest flux rates) to 3 (lowest flux rates). The intensity of the orange and green bars on the top and side represent 5-deoxyribose exchange rate and biomass production, respectively. The rate of both these fluxes can be partitioned and can be used with high confidence to provide clear distinctions between the clusters of conditions. The partitioning process we used was able to provide a similarly clear distinction in both dimensions using each of the fluxes reported in Table 5.1.



(a) Coarse clustering. **(b)** Fine clustering.

Figure 5.3: Heat maps of the similarity matrices of the case study 2 dataset, fused by subsystem and arranged by spectral clustering. The x and y axes represent the samples generated by the simulation, whilst the intensity of the colours in the centre represent the similarity between each of the pairs of x and y samples. The intensity of the blue and green bars on the top and side represent glucose exchange rate and biomass production, respectively.



(a) Experimental data from gene expression. **(b)** Simulated data from case study 2 dataset.

Figure 5.4: Heat maps of the similarity matrices from both experimental and simulated data, with their rows and columns organised by biomass and Fe^{2+} fluxes respectively. The x and y axes represent the samples generated by the simulation, whilst the intensity of the colours in the centre represent the similarity between each of the pairs of x and y samples. The intensity of the blue and green bars on the top and side represent Fe^{2+} exchange rate and biomass production, respectively. These heat maps clearly show distinct phases of correlated and uncorrelated behaviour between the fluxes.

CHAPTER 6

Hierarchical Block Matrices and Local Network Learning

6.1 Introduction

Hierarchical Block Matrices, or HBMs [165], are a structure of similarity matrix, which can be recovered by performing iterative Markov clustering to aggregate the data into a tree structure [166]. This is highly applicable to metabolic network analysis as a way to identify subsystems at a variety of levels of organisation. For this reason, I discuss here my applications of HBMs first to simple network structure measures that are agnostic of the underlying network type, and then on to more advanced measures based on similarities of node fluxes.

A novel similarity estimation approach, termed Local Network Learning, was used to approximate conditional dependencies between network nodes, exploiting the flow network structure itself in order to remain computationally tractable even on large datasets.

The aim of this chapter is to introduce the use of Hierarchical Block Matrices and Local Network Learning to detect network structure in datasets of similar metabolic flow networks, with the hypothesis that this combination of methods can uncover meaningful data about network structure. Specifically, this approach is evaluated by comparison with existing, manually curated subsystem labels, and found to outperform more traditional methods.

The work in this chapter was presented at BBCC2018.

6.2 Testing on synthetic data

Hierarchical Block Matrices have been used in generative contexts a number of times [167–169] for testing hierarchical graph clustering algorithms, but the approach of using an

explicit iterative Markov clustering approach to reconstruct Hierarchical Block Matrices directly has only previously been used in the context of chromatin contact clustering [166]. Given the quite different context in which they will be used in this chapter, it is important to demonstrate the usage of HBM based structure detection on synthetic graphs. In general, in order to validate the effectiveness of an unsupervised learning technique, it is important to first test it on data with known 'correct' answers. Since this method is intended to find hierarchical structure in network data, synthetic datasets are needed which both definitely possess such structure, and definitely do not.

Figure 6.1 shows the results of these tests.

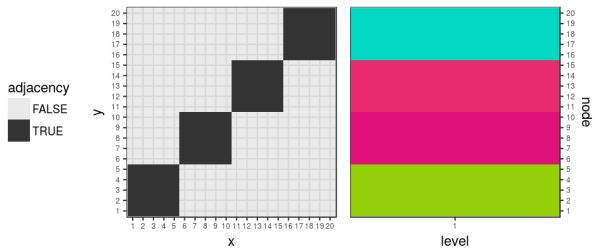
The first tests, shown in figure 6.1a, figure 6.1b, figure 6.1c and figure 6.1d show successful detection of structure in graphs with hierarchical structure, whilst figure 6.1e shows a true negative result in a completely connected graph. Figure 6.1f and figure 6.1g show slight true positives, where the algorithm divides the lattice and ring graphs in half. Finally, figure 6.1h shows a successful detection of the major structure in an environment with some noise. In the figure, noise was introduced by overlaying a constant lattice for the sake of repeatability, but similar results are also found using normal and uniform noise.

In figure 6.1, the inflation parameter to the Markov clustering subroutine was fixed at 3 in every test, in order to enable clear comparison. This is higher than the default value of 2, since from experimentation it appears that in this context a higher inflation parameter leads to the identification of more levels of hierarchy (as shown in figure 6.1b and figure 6.1d), at the expense of the potential for true positives such as in figure 6.1f and figure 6.1g. The same value of 3 is used throughout the rest of this chapter, since experimentally it appears to be high enough to find large amounts of structure, without giving too many false positive levels of hierarchy when used on real data, as evidenced by cluster stability across small network perturbations.

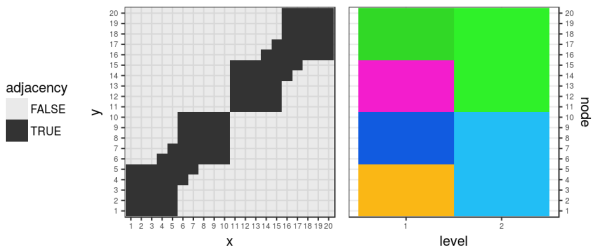
These results demonstrate the ability of Hierarchical Block Matrices to detect structure in synthetic graphs, showing that their use for the same task on real metabolic networks is at least feasible. These synthetic graphs have been introduced as adjacency matrices with an obvious notion of similarity between nodes, but in real world networks the best notion of similarity is less clear, as will be discussed in the next section.

6.3 Network structure measures

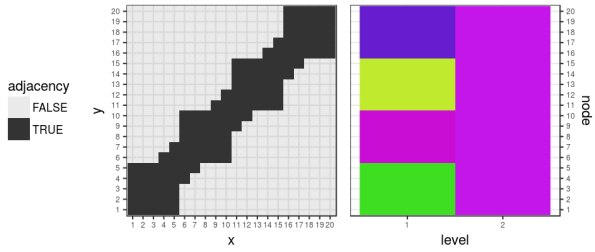
Hierarchical Block Matrices have found use in chromatin contact maps, where the input data is a boolean specifying whether a pair of nodes interact. This means that a logical starting point is to base the structure measure on the existence of a link between two nodes in question. However, as discussed in section 2.7.3, metabolic networks, like many



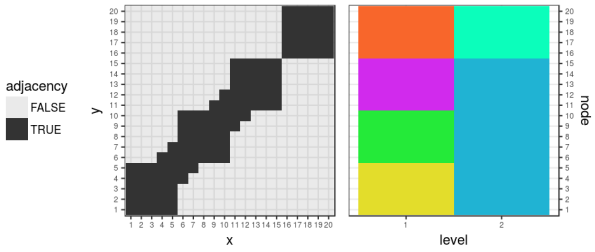
(a) Four cliques.



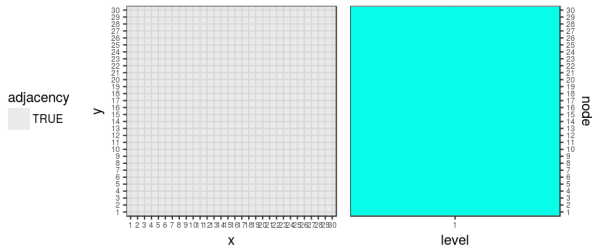
(b) Four cliques, joined in pairs.



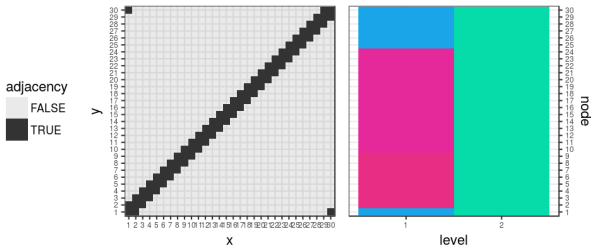
(c) Four cliques joined in a line.



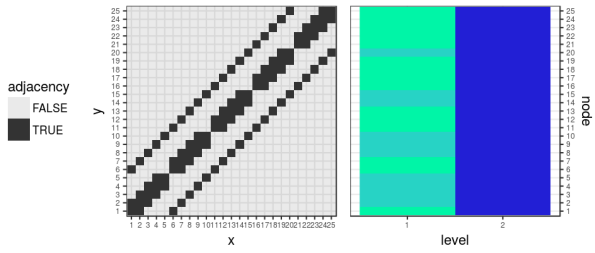
(d) Four cliques, three of which are joined in a line.



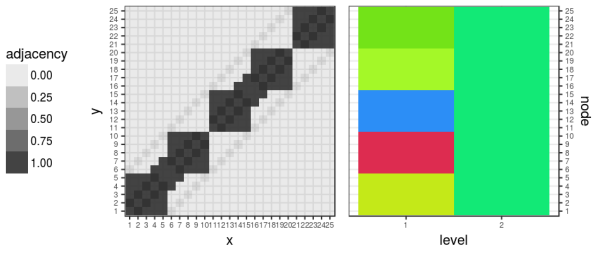
(e) Complete graph.



(f) Ring.



(g) Lattice.



(h) Four cliques joined in pairs, plus noise.

Figure 6.1: Plots showing validation of HBM usage for graph clustering. Left hand sides of plots show adjacency matrices, right hand sides show the resulting hierarchical clusters.

kinds of flow networks, are naturally described by bipartite graphs. These have two classes of nodes: metabolite nodes, which describe physical entities in the system, and reactions, which describe processes transforming between these entities. This means that it does not make sense to use direct links, but instead to summarise the bipartite graph as two boolean adjacency matrices, one for reactions and one for metabolites, with the reaction matrix being TRUE where the two reactions share a metabolite, and FALSE otherwise, and vice versa for the metabolite matrix. This can be calculated as the reaction or metabolite subgraph of the square of the adjacency matrix.

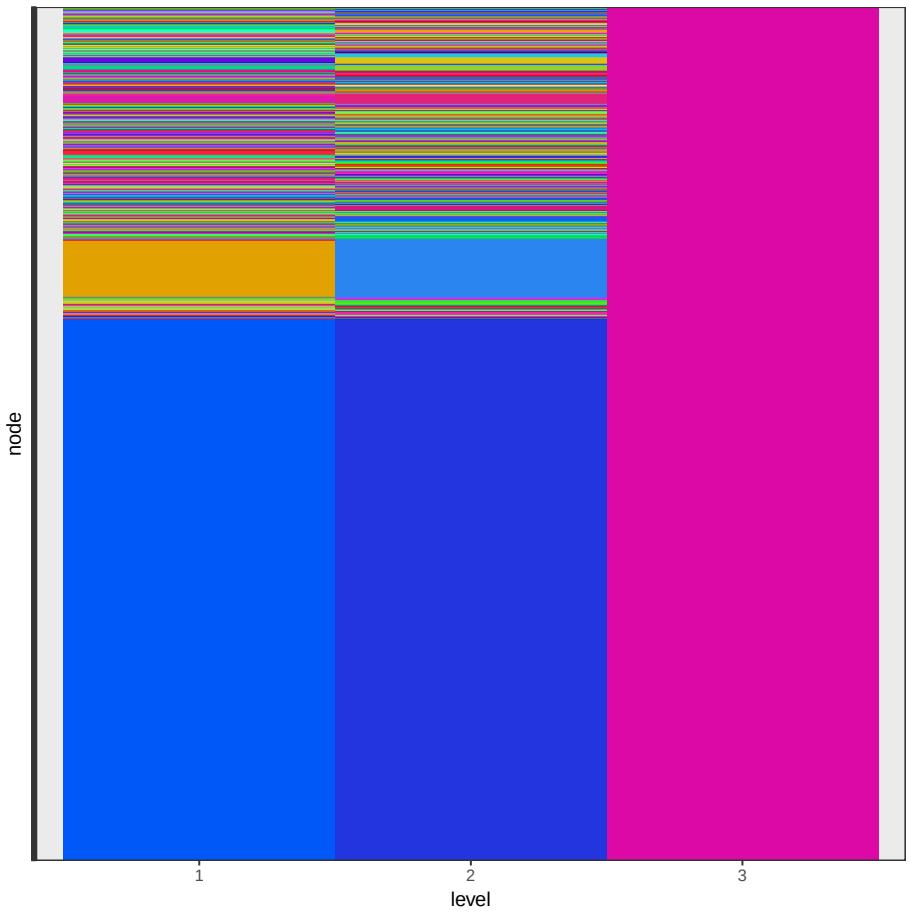


Figure 6.2: Plot of clustering of reactions provided by Hierarchical Block Matrix with boolean links based on metabolite sharing. Y-axis shows reactions (labels are omitted for clarity). X-axis shows hierarchical levels. Colours represent clusters, but are arbitrarily chosen so matching colours only indicate matching clusters within columns.

Creating a Hierarchical Block Matrix using just boolean reaction connectedness already shows some promise—in figure 6.2, it is possible to see the identification of two main clusters, one large and one small, along with a large number of unclustered reactions, shown by the areas with narrow colour bands. However, this clustering does not appear to have identified any hierarchical structure, and the coarse nature of the clustering, particularly the grouping of the highly connected core as a single cluster covering

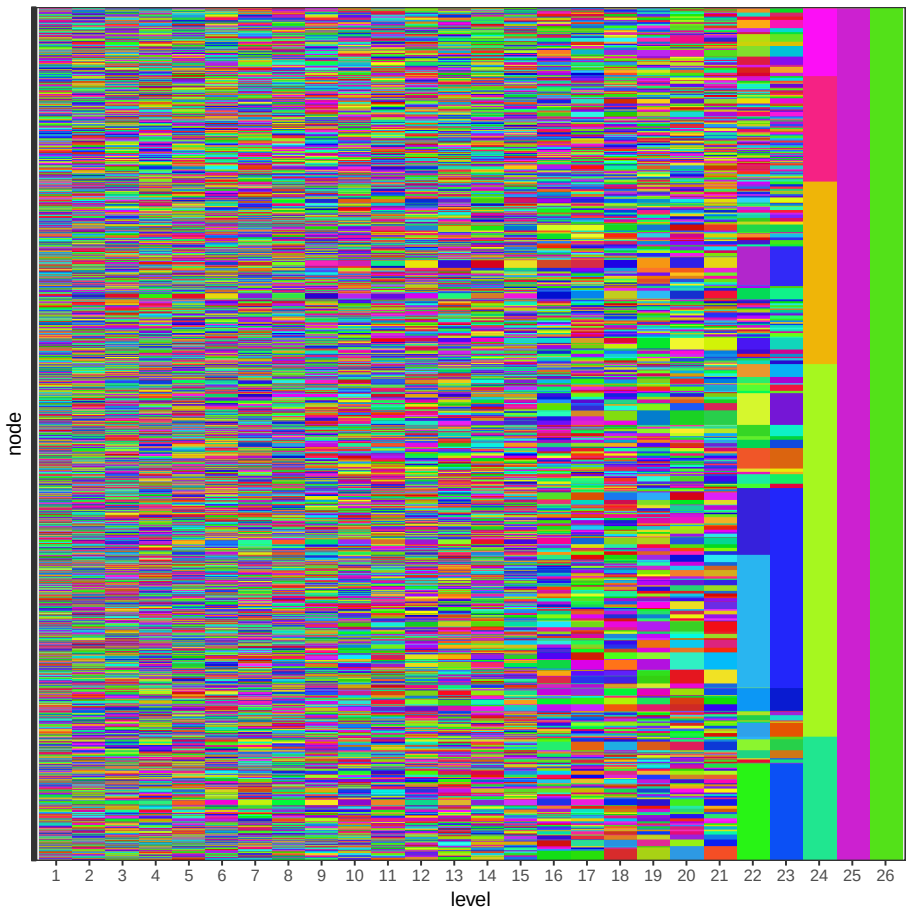


Figure 6.3: Plot of clustering of reactions provided by Hierarchical Block Matrix with boolean links based on inverse connectivity of shared metabolites. Y-axis shows reactions (labels are omitted for clarity). X-axis shows hierarchical levels. Colours represent clusters, but are arbitrarily chosen so matching colours only indicate matching clusters within columns.

over half the nodes, make this result only useful as a proof of concept.

One approach to improving the effectiveness of this approach in the highly connected core is to attempt to highlight the most important connections. In figure 6.3 this is achieved by setting connection strength between two reactions via a metabolite m to equal $1/deg(m)$, or vice versa for metabolites. This is very successful: it yields many more levels, and it also shows hierarchical structure. This structure is particularly obvious in the highest levels, but a more detailed examination in fact finds that structure is discernible on around 10 levels. This is promising, since it shows that the core concept of HBM clustering to identify hierarchical metabolic network structure is effective, if used with the correct reaction similarity measure.

These connectivity based approaches have the advantage that they are not really specific to metabolic networks at all, since they involve no simulation; this means that they can be applied to a variety of areas, with potentially different network measures applicable in each situation. For instance network distance is more applicable to situations with a

more clear idea of an additive link cost, such as latency in communications networks.

However, using measures of node similarity that are based purely on network structure is somewhat unsatisfying, precisely because they can be applied to pretty much anything that can be represented as a network. It would be superior to have measures that are more specific to the semantics of flow networks, and which can capture the possibility that different subsystem descriptions are appropriate under different conditions.

6.4 Local Network Learning based similarity measures

6.4.1 Predictivity gain

In order to create a measure of node similarity that uses more of the semantics of the network, we can use a similarity between the fluxes in the nodes to indicate to what extent nodes belong together. Correlation based flux similarity measurements such as those discussed at other points in this thesis are simple to understand and implement, but they are limited in terms of the kinds of relationships they can pick up on. It would be preferable to use more flexible models to capture the strength of the relationship between two fluxes by using an 'information gain' style approach [170].

To describe this predictivity gain approach, it is best to first define some variables while initially ignoring the network structure of the problem:

$$\begin{aligned}
\mathbf{B} &\coloneqq \text{a matrix with } p \text{ rows and } q \text{ columns}, && (6.1)\\
T(\mathbf{y}, \mathbf{X}) &\coloneqq \text{a training function, where} && (6.2)\\
&\quad \text{height}(\mathbf{X}) = \text{length}(\mathbf{y}), && (6.3)\\
&\quad \text{and returning a prediction function} P(), && (6.4)\\
P(\mathbf{X}'), &\coloneqq \text{a prediction function, which returns a vector } \mathbf{y}', \text{ where} && (6.5)\\
&\quad \text{length}(\mathbf{y}') = \text{height}(\mathbf{X}'), && (6.6)\\
C(\mathbf{y}, \mathbf{y}'), &\coloneqq \text{a cost function, which} && (6.7)\\
&\quad \text{takes two equal length vectors, and} && (6.8)\\
&\quad \text{returns a scalar measure of their dissimilarity, with} && (6.9)\\
&\quad \forall \mathbf{a}, C(\mathbf{a}, \mathbf{a}) = 0 && (6.10)
\end{aligned}$$

We can then define the predictivity gain, G, provided by a column $\mathbf{a}$ on another column, $\mathbf{b}$, both of matrix $\mathbf{B}$ as follows, using the notation $\mathbf{B}\backslash\mathbf{a}$ to mean 'matrix $\mathbf{B}$,

except for column $\mathbf{a}$'.

$$G(a,b) = C(\mathbf{b}, T(\mathbf{b}, \mathbf{B})(\mathbf{B}\backslash\mathbf{b})) - C(\mathbf{b}, T(\mathbf{b}, \mathbf{B}\backslash\mathbf{a})(\mathbf{B}\backslash\mathbf{b}\backslash\mathbf{a})) \tag{6.11}$$

For simplicity, this equation assumes that the training function T is perfectly regularised—in real life, cross validation is normally required.

However, there are some problems introduced by this approach. Firstly, when the number of available features q is large, training a model takes a long time and requires a large amount of data and a highly regularised model to avoid overfitting. Secondly, we may have a situation where more than one feature in $\mathbf{B}\backslash\mathbf{b}$ is a very good predictor of $\mathbf{b}$. In this case, we would end up assigning a low similarity to each, since although they might all be able to deliver a high quality prediction on their own, they would all deliver little information gain over and above the others. The best way to remedy this would seem to be to try every combination of features, and weight the importance of an individual feature based on the predictive power of all combinations that it was used in. However, this approach increases the computation time with $\mathcal{O}(q!)$.

This line of reasoning takes us to the idea that this approach could be infeasible with a large number of nodes, as we see in metabolic networks. However, the fact that we are dealing with a network in fact allows us to overcome this difficulty.

6.4.2 Predictivity gain applied to metabolic networks

In order to apply a predictivity gain approach to a set $\mathbf{S}$ of n flux vectors obtained from n metabolic networks using the terminology of equation (2.1), we can use the following assignments:

$$p = n \tag{6.12}$$

$$q = r \tag{6.13}$$

$$\mathbf{B} = [\{\mathbf{x} : \mathbf{x} \in \mathbf{S}\}]^{\mathsf{T}} \tag{6.14}$$

In other words, we form the matrix $\mathbf{B}$ by vertically concatenating the flux vectors obtained by maximising each of a set of metabolic networks. This means that the ordering of the rows and columns in $\mathbf{B}$ is unimportant, unless the original set of metabolic networks, or their reactions were ordered. However, while unordered, the columns of $\mathbf{B}$ do possess important structure. Since each column corresponds to a particular reaction, they have an implied network distance which we can exploit.

In order to effectively explain this, we need to introduce some more definitions:

$$G \coloneqq \text{ the metabolic network graph, with adjacency matrix } \begin{bmatrix} 0_{r,r} & \mathbf{A} \\ \mathbf{A}^\mathsf{T} & 0_{m,m} \end{bmatrix} \neq 0 \tag{6.15}$$

$$R \coloneqq \text{ the reaction adjacency graph, with adjacency matrix } \mathbf{A}\mathbf{A}^\mathsf{T} \neq 0 \tag{6.16}$$

$$N_R(v) \coloneqq \text{ the function which returns the set of vertices which are adjacent to node } v \text{ in graph } R. \tag{6.17}$$

It is important to note here that $N_R(v) = N_G(N_G(v))$, due to the bipartite nature of metabolic network graphs such as G, . In other words, adjacent reactions in R are not directly connected in G, but instead are connected at a distance of 1, which means that they share at least one metabolite.

With these definitions, for any given reaction v_0, the neighbouring reactions $N_R(v_0)$ (for example $v_{1\ldots4}$ in figure 6.4) can be thought of like a Markov blanket. In other words, the rate of reaction v_0 can only be directly dependent upon the rates of reactions in $N_R(v_0)$, since $N_R(v_0)$ are the set of reactions that involve any metabolite which is involved in v_0; any other reactions can only constrain v_0 indirectly, via constraints on reactions in $N_R(v_0)$. This means that we can fully estimate the degree of control of reaction v_0 only with reference to reactions $N_R(v_0)$, which is a much smaller set, and therefore allows us to take this nonlinear, information gain based approach.

6.4.3 Calculating predictivity

At this stage, we wish to estimate the importance of the rate effect of each neighbouring reaction to a reaction of interest, by comparing the effectiveness of models with and without the neighbouring reaction. In order to do this we can take a supervised learning approach, regarding each neighbouring reaction as a feature and the target reaction as our labelled data. This means that we can consider this approach as constructing a feature selection problem for each node, specifically with a backward elimination style approach [171]. This is an appropriate approach because the goal here is to exhaustively and fairly evaluate feature quality, rather than taking a heuristic approach to get the best possible prediction as quickly as possible. It is worth comparing this strategy, which involves retraining a model with and without each feature to an input fuzzing approach, which would involve training one model and then replacing each feature in turn with disinformation, such as zeros or random noise. The problem with a fuzzing approach is that it implicitly relies on the trained model having no internal biases between features. For instance in the case of two identical features, we would need to rely on the trained model having treated both identically, rather than just using the first encountered and

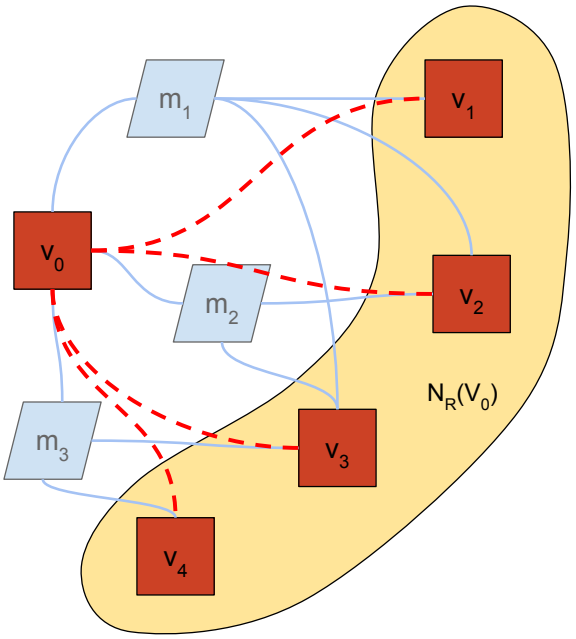


Figure 6.4: A sketch demonstrating reaction neighbourhoods. In this sketch, red nodes represent reactions, light blue nodes represent metabolites, and light blue lines represent their interactions. Red dashed lines represent reaction adjacency, and the yellow highlighted area shows the results of the reaction adjacency function.

discarding the other.

From here on, the term "Local Network Learning" to refer to this approach of quantifying edge importance by how much information gain it provides on a node's value. A literature search found that this technique appears to be novel in general, and is certainly novel in this context. The most similar approaches are regulatory and Bayesian network reconstruction techniques, which are similar in their domain and data types; and wrapper methods of feature selection, which have a similar intention and technical approach, apply to tabular data.

6.4.4 Choice of supervised learning algorithm

The next question is what class of supervised learning algorithms to use. I have intentionally designed this approach to yield a large number of well structured tabular problems, rather than any more exotic data type, since this opens up a wider range of candidate algorithms. However, because the number of problems created is very large, the technique chosen must be fast to train. Decision tree models are an obvious candidate here because they are fast to train and capable of modelling the kind of piecewise linear prediction landscape that is expected from this problem—as mentioned in Chapter 2, the feasible space of linear programming problems is a complex polytope. They work well as a coun-

terpoint to the use of pure regression models, which have the advantage that they are capable of not displaying feature bias, in the sense described in the last paragraph, and have demonstrated their effectiveness at other points in this thesis.

6.4.5 Applying network local predictive power based similarity measures

Demonstrating a Local Network Learning approach requires a set of test datasets, and the selection of examples of regression algorithms to use as the kernel that estimates node values from their neighbours.

The datasets described in section 3.4 were created with the testing of a Local Network Learning approach in mind, and are used here as demonstration data. This is one of the reasons for the particular focus on enhancing the variance in the dataset, since if the value of a node is constant, it is not possible to measure the effects of the adjacent nodes. Even when nodes are not constant, if the value of too many nodes are too easily predicted, then too many perfect models are produced, making the resulting dataset of model errors too sparse and difficult to work with.

Although using a local modelling technique cuts the computational load significantly (see section 6.4.6), there are still a large number of models to be created, meaning that performance was a primary concern in the choice of regression algorithms.

The two algorithms selected were:

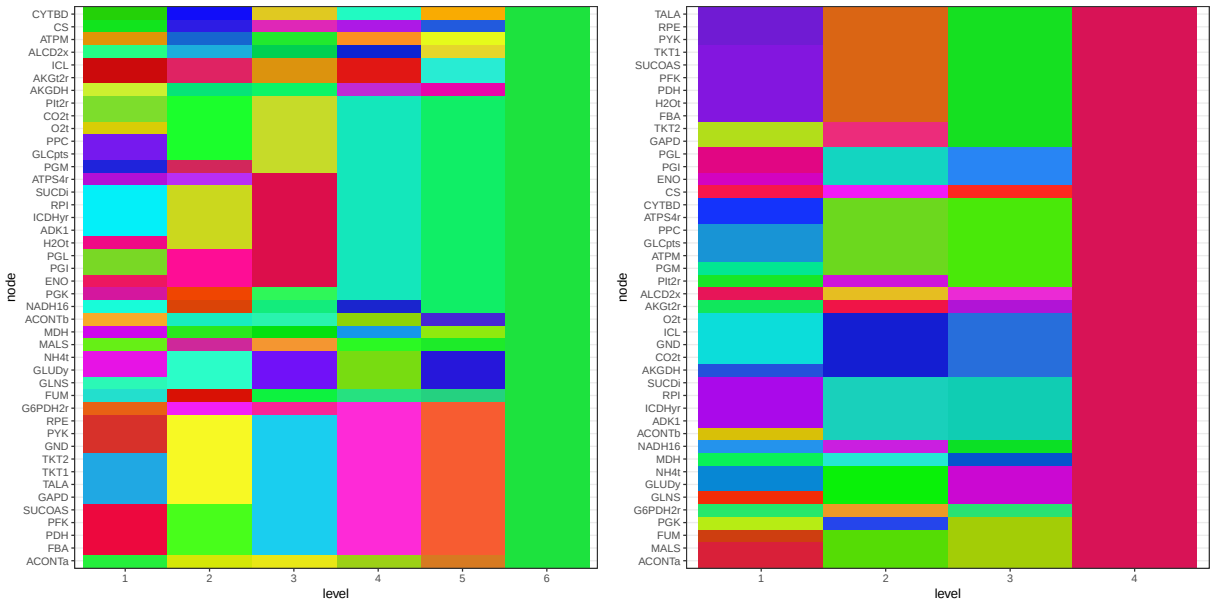
- simple linear regression, for its speed and for the correspondence with the base case of using correlation as a similarity estimator, and
- the Cubist regression tree algorithm [172], for its speed and because I had previously found it to have very good performance in this type of application [12].

6.4.6 Reducing computational load by enhancing graph sparsity

This network local predictivity approach requires the construction of $\mathcal{O}(|E|)$ models, where $(|E|)$ is the number of edges in the graph, with the mean number of features of the models proportional to the mean degree of the graph. The time to build a decision tree scales with at least $\mathcal{O}(m)$ in the number of features, but more commonly $\mathcal{O}(m^2)$, since without making strong independence assumptions, every feature must be compared to choose every split [173]. This gives an overall complexity of $\mathcal{O}(|E|^3)$, but note that the relevant graph here is a graph of connections between reactions. To construct this we must remove each metabolite node and add edges connecting every neighbour of each removed node. Obviously this operation significantly increases the edge count, and with it the time complexity, particularly for those metabolites with large numbers of edges.

This increase in edge count was a substantial problem for large models. Fortunately, however, the structure of the reaction networks made it possible to increase sparsity with minimal loss of information. The number of reactions per metabolite is highly skewed, with a 95th percentile of 10, but a maximum of over 1000. This meant that 95 % of edges could be removed from the graph by removing only around 0.5 % of metabolites, giving a speedup on the order of 8000 fold. The metabolites removed by this approach were manually checked to ensure that they were biologically reasonable to ignore, such as water and oxygen, which can typically be assumed to be available in excess, and so were not constraining the rates of reactions that consume them.

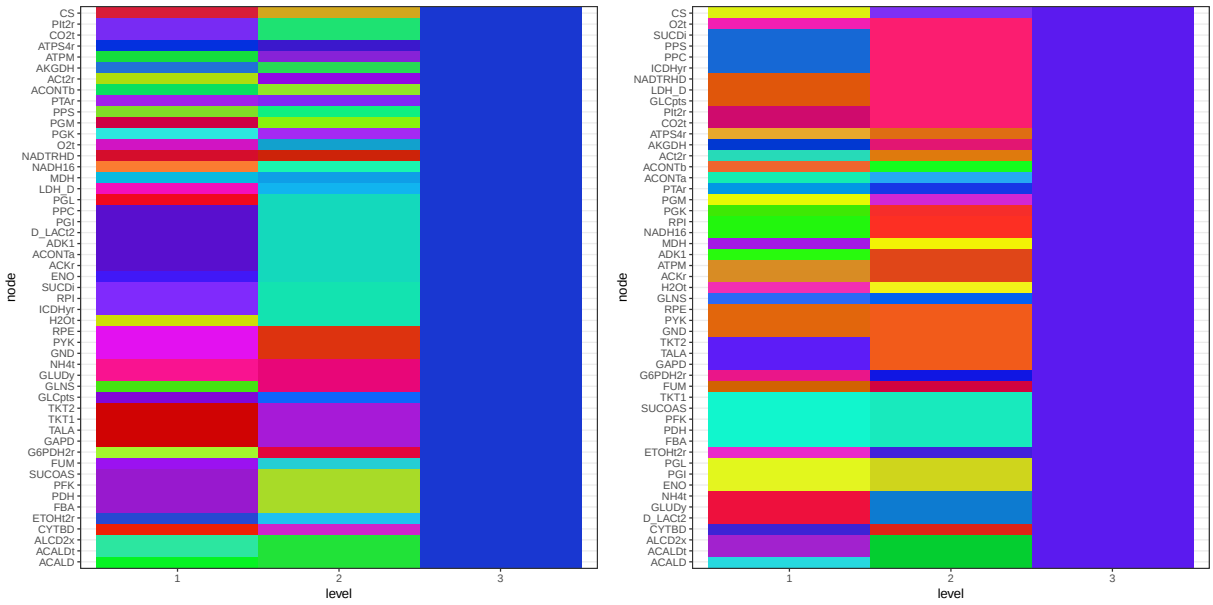
Another performance advantage of this method is derived secondarily from this reduction in the typical number of features per model. When one is training models with large numbers of features, it is necessary to use large sample sizes to avoid overfitting. However, in this case the local learning approach already selects features down to a smaller set which are expected to have a large correlation with result, so it is possible to use a smaller sample size than would otherwise be required. For decision trees, execution time scales with $\mathcal{O}(n)$ in the number of samples, so the execution time reduction was relatively modest, but the reduction in memory requirement, also $\mathcal{O}(n)$, was far more significant.



(a) Linear model. **(b)** Cubist regression tree.

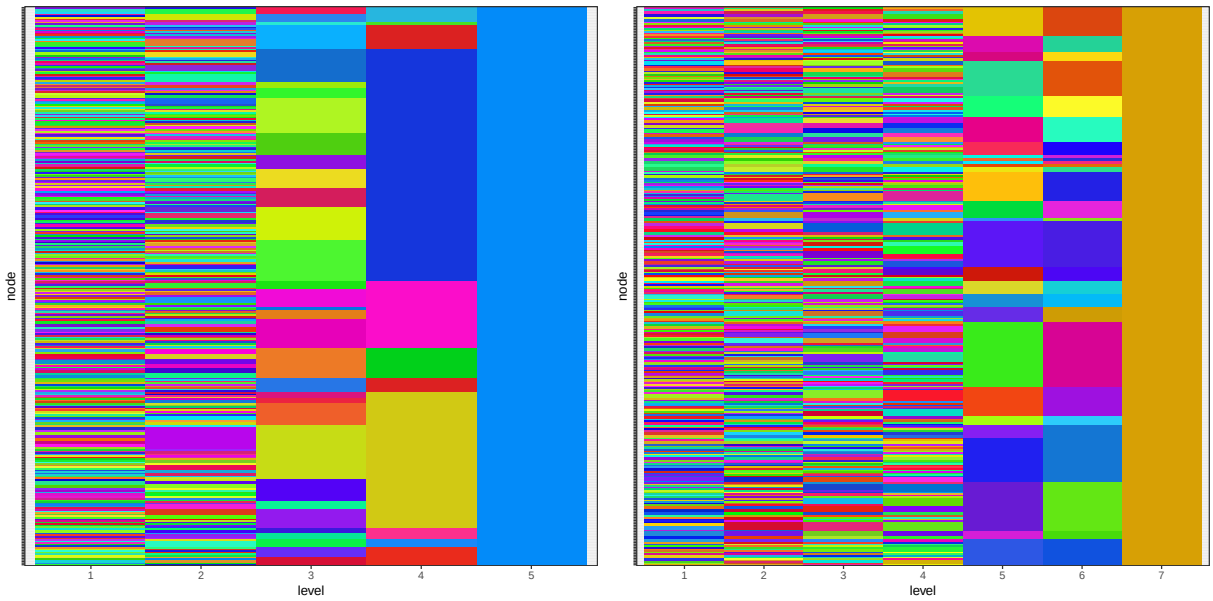
Figure 6.5: Plot of reactions (y-axis) against clustering levels (x-axis) for case study 0 dataset (placeholder sampling approach). Colours represent clusters, but are arbitrarily chosen so matching colours only indicate matching clusters within columns.

In figures 6.5, 6.6 and 6.7, we can see that in most cases this technique produces high quality hierarchical clusterings across a wide range of simulated datasets without requiring significant tuning, and whilst using two very different regression algorithms to conduct local learning. Figure 6.8 shows that the success of this clustering approach



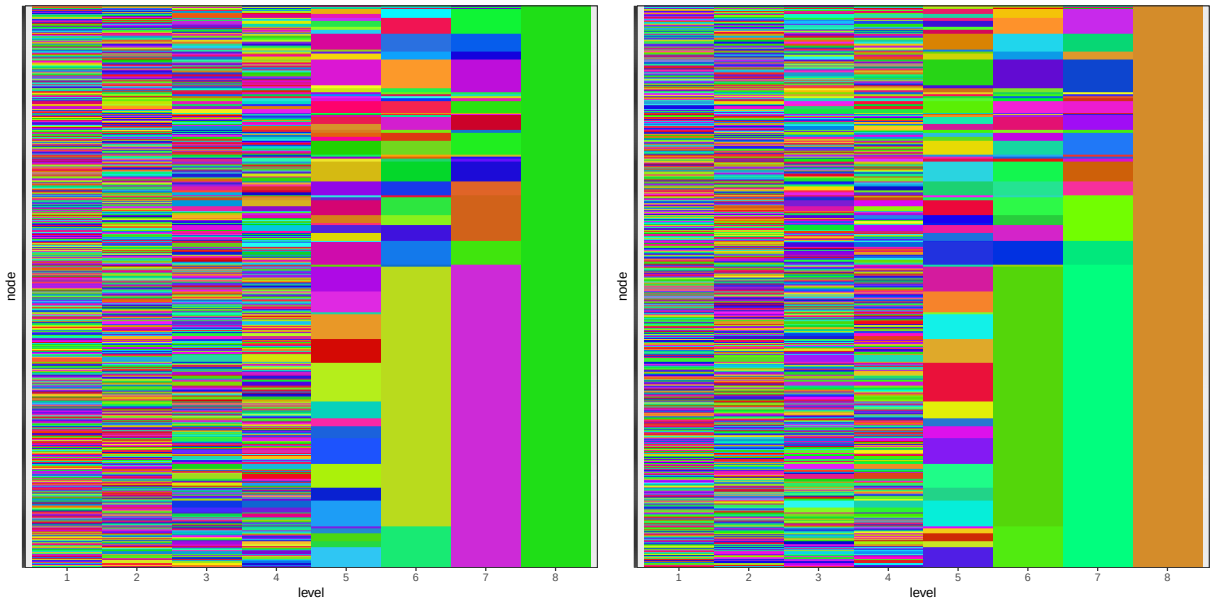
(a) Linear model. **(b)** Cubist regression tree.

Figure 6.6: Plot of reactions (y-axis) against clustering levels (x-axis) for case study 1 dataset (batch based proportional adjustment). Colours represent clusters, but are arbitrarily chosen so matching colours only indicate matching clusters within columns.



(a) Linear model. **(b)** Cubist regression tree.

Figure 6.7: Plot of reactions (y-axis) against clustering levels (x-axis) for case study 2 dataset (evolutionary sampling with environment variation). Colours represent clusters, but are arbitrarily chosen so matching colours only indicate matching clusters within columns.



(a) Linear model. **(b)** Cubist regression tree.

Figure 6.8: Plot of reactions (y-axis) against clustering levels (x-axis) for a data set derived from Colombos database of real world gene expression, similar to that used in Chapter 5. Colours represent clusters, but are arbitrarily chosen so matching colours only indicate matching clusters within columns.

carries over to use on real world data; furthermore this was possible with only a handful of changes to normalisation code. This confirms that this technique can be used as a structure detection approach seamlessly for both theoretical analysis of metabolic models themselves and applied analysis of the predictions of those models on real world data. Since this approach was successful even with the prokaryotic methods used here, it is likely that it would show even greater success on larger, eukaryotic models, since these have organelles which display further inherent structure.

The clusters found show consistency in some areas across different datasets and techniques, whilst also showing the differences that should be expected given the differences in the underlying datasets and prediction algorithms. The hierarchical cluster structure is supported by manually curated subsystem labellings available for the underlying *E. coli* model, whilst also providing hierarchical structure that clusters both more and less coarsely than these manual labellings. Finally these clusterings are condition dependent: we can see how they vary depending on the flow patterns in the dataset, providing valuable information about condition dependent hierarchical structure.

6.5 Validation

As stated previously, validation of clustering is challenging, especially when the goal is knowledge discovery, rather than the replication of already known structure. However, in

this situation, we are lucky enough to have at least one known true assignment of reactions to clusters: the subsystem labels available in the dataset. This is not the only possible assignment by any means, but nevertheless we would hope that HBM based clustering shows at least some similarity with the subsystem assignment.

In order to show this relationship, the Adjusted Rand Index [174] was calculated for each clustering level in each Hierarchical Block Matrix against the human derived subsystem assignments. I also calculated the Adjusted Rand Index for multiple shuffled versions of each of the HBM assignments (in blue), in order to provide a baseline, since although the Adjusted Rand Index is already corrected for the expected similarities between datasets by chance, providing an experimental counterpoint also provides us with a way to see the variability. Trivial clusterings with only one cluster were filtered out.

This validation process was partly inspired by that used in Zhao *et al.*[175], which also examines network structure and uses a shuffled baseline, although the baseline is generated by shuffling the network rather than the labelling, as here. In addition, I evaluate the results via comparison with human labelled subsystems, whereas in Zhao *et al.*, the comparison is purely derived from network measures.

Figure 6.9 shows the results of this validation—red box plots represent the various levels of HBM cluster assignments found, while the blue box plots represent the corresponding shuffled cluster assignments. We can see clearly that the Hierarchical Block Matrix with Local Network Learning approach outperforms the shuffled variant, and this is borne out statistically: $p < 0.05$ for 5 out of the 8 groups.

The final three rows in figure 6.9 represent more conventional clustering approaches for comparison, showing that my approach generally outperforms them. K-means was used in the context of clustering reactions directly by their flux values, while the walktrap algorithm was used to cluster based on the network structure itself.

The Adjusted Rand Index is lower for the biological model, based on gene expression data. A significant reason for this is that this dataset had both smaller variance in reaction rates and a smaller sample size than the synthetic datasets, demonstrating the utility of synthetic dataset generation.

6.6 Conclusion

This chapter demonstrates for the first time the utility of Hierarchical Block Matrix techniques for the detection of hierarchical structure in metabolic networks. This is shown initially using network structure based similarity measures, before introducing a Local Network Learning approach to efficiently estimate predictivity based similarity measures between flow network nodes. The Local Network Learning approach introduced appears to be novel in the context of metabolic networks, certainly with the implementation

described here, and may also be novel in a wider context.

The combination of hierarchical block matrices and Local Network Learning was also able to extract hierarchical structure from metabolic network datasets, and this structure achieved an Adjusted Rand Index against existing manual labellings of reactions that was both statistically significant and superior to what which was found using comparable methods.

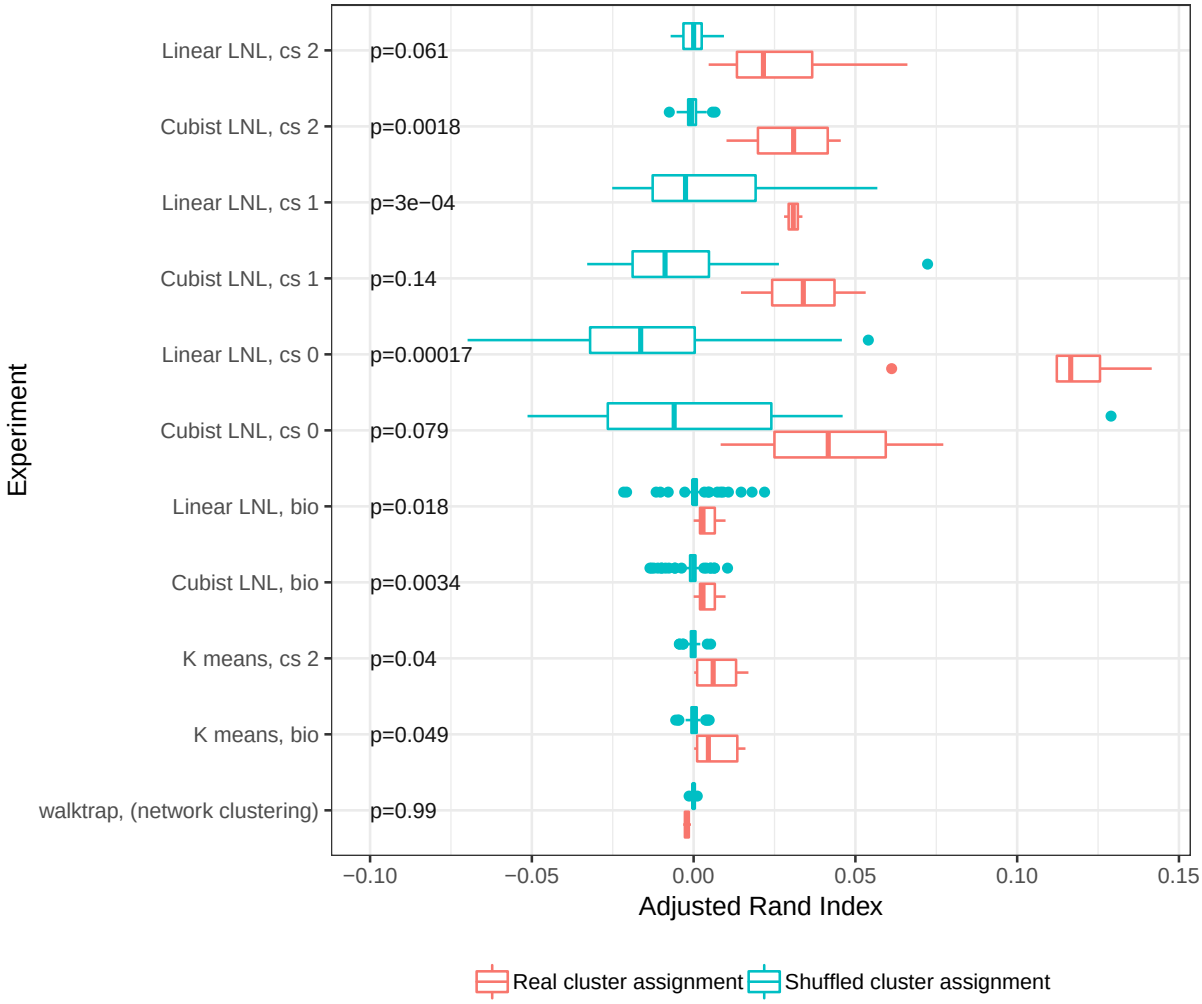


Figure 6.9: Adjusted Rand Index values compared to human labelled subsystems for real (red) and shuffled (blue) Hierarchical Block Matrix clusters. All box plots show Adjusted Rand Index versus human labelled subsystems, across different clustering levels. The x axis shows Adjusted Rand Index. The y axis shows experiment groups: combinations of a machine learning method (linear model or Cubist decision tree) used for local network learning and a dataset to which it was applied: either a simulated case study or biological gene expression data. P-values for each comparison are shown in black. The final 3 rows are controls provided by k-means and walktrap clustering algorithms.

CHAPTER 7

Future work

7.1 Network of networks *vs* structure of solution spaces

In this thesis, the term 'structure is used in two slightly different contexts. There is structure as used in Chapter 6, that is the structure of the metabolic network itself, such as pathways and subsystems. There is also structure as described in Chapter 5, which is the structure of the solution space of the metabolic model, such as clusters of solutions that behave similarly. These concepts are both interesting and important, but can be thought of as conceptually perpendicular—they are much like the difference between grouping a table by rows or by columns. An obvious future direction for this research would be towards methods that can detect structure in both directions at once. The most obvious method for this would be to work in one direction and then the other, perhaps by finding groups of solutions and then separately identifying network structure within these groups. However, this would mean that no information was transferred between the network structures of different groups, so an ideal solution would instead somehow perform the grouping in both directions simultaneously.

7.2 Precomputing datasets for FBAonline

As discussed in section 4.4, at present, the FBAonline platform is limited by its web application nature to using those network exploration techniques that run in near real time. However, this situation could be improved by noting that while a large number of metabolic network models exist, the number of high quality, human verified models that are readily available from online databases numbers only in the hundreds or low thousands, meaning that it would be tractable to precompute some analysis of these. This approach would be particularly appropriate for the local network learning technique described in Chapter 6, where calculating the distance estimate between nodes is much

slower than the analysis of the resulting distance network.

7.3 Other flow network data

Although the focus of this thesis is metabolic networks, much of the content is in fact agnostic to the data type, and similar techniques could be applied with minimal modification to other fields.

Potential examples include infrastructure networks, such as road and logistical networks, and distribution systems for utilities such as water and electricity, as well as information networks which exhibit some conservation of flow, such as packet switched communications networks and financial transaction networks.

CHAPTER 8

Conclusion

This thesis has explored methodology for the detection and interpretation of structure in metabolic flow networks, using statistical analyses over large datasets.

This began with Chapter 3, which introduced the concept of biased Monte-Carlo sampling over metabolic network state spaces, and described how this idea could be used as the core of a conceptual framework to allow the same analysis techniques to be used seamlessly in contexts from pure *in silico* simulation to multi-level integration of real *in vivo* data. Chapter 3 also catalogued the primitives that could be used to build data sets from either biological data or Monte-Carlo simulations, and demonstrated these primitives in the creation of three case study datasets of varying complexity. Finally, section 3.4.1 introduced a popular new software tool for conducting these simulations, which is faster than any other tool tested. This allowed two of these case studies to be what appear to be the largest datasets of their type, at half a million samples each.

Chapter 4 is a description of three research projects which worked with datasets which were small enough for the use of analytic techniques that involved visualizing flux data directly. The first of these discusses parameter tuning a metabolic model of Salmonella in order to fit experimental gene expression values. This provided a mechanistic explanation of growth differences between conditions which were separately identified experimentally [9]. The second of these projects continued a theme of comparative metabolic modelling, by comparing datasets created by evolutionary sampling of two metabolic models of different *Geobacter* species. This was successful in providing clear explanations of differences between the species [10], but an attempt to provide an easily useable platform to repeat this analysis was less successful. The third project in this chapter addressed this in the creation of a much more usable tool for metabolic network comparison, which was then used in the creation of a new organism model [11]. These projects also described different approaches to visualizing differences between metabolic networks. While these visualisation approaches had various strengths, their common weakness was the inability to scale

to whole organism networks, demonstrating the need for the statistical techniques in the later chapters. These projects (and the case studies described in Chapter 3) involved the creation of three new software tools: the visualization tools Metabex and FBAonline, and a core metabolic network simulation and manipulation library, Fbar.

Chapter 5 shows the use of a modified version of the Similarity Network Fusion algorithm to a large metabolic and gene expression dataset, along with a simulated dataset created in Chapter 3, resulting in the detection of quantised cell states in both. These quantised states were also shown to be linked to similar reactions in both, demonstrating that they were an emergent product of the metabolic network structure. This represented the first use of Similarity Network Fusion, with or without modification, in a multi-level metabolic context, and successfully showed metabolic network structure across multiple levels in real world data.

Finally, Chapter 6 demonstrates for the first time the ability of Hierarchical Block Matrix clustering approaches to detect structure in metabolic networks, and introduced the novel Local Network Learning similarity calculation approach. This created a technique for hierarchical network structure detection which was validated against manually labelled subsystems and found to outperform both a random baseline and a selection of standard clustering algorithms.

Overall, this thesis shows that structure can be detected in metabolic networks by using machine learning approaches on large sets of feasible metabolic network states, and introduces a number of new methods for this. This is shown in both the network structure itself and the structure of populations that is implied by the networks themselves. Furthermore, a large advantage over purely analytic approaches is shown: using machine learning methods allows for a seamless transition between theoretical testing using biased Monte-Carlo simulations and practical inference on real world data.

Bibliography

[1] Amit Varma and Bernhard O Palsson. Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type escherichia coli w3110. *Appl. Environ. Microbiol.*, 60(10):3724–3731, 1994.

[2] Kiran Raosaheb Patil, Mats Åkesson, and Jens Nielsen. Use of genome-scale microbial models for metabolic engineering. *Current opinion in biotechnology*, 15(1):64–69, 2004.

[3] Zachary A King, Justin Lu, Andreas Dräger, Philip Miller, Stephen Federowicz, Joshua A Lerman, Ali Ebrahim, Bernhard O Palsson, and Nathan E Lewis. Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic acids research*, 44(D1):D515–D522, 2015.

[4] Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

[5] Natapol Pornputtapong, Intawat Nookaew, and Jens Nielsen. Human metabolic atlas: an online resource for human metabolism. *Database*, 2015, 2015.

[6] Bo Wang, Aziz M Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature methods*, 11(3):333, 2014.

[7] Supreeta Vijayakumar, Max Conway, Pietro Lió, and Claudio Angione. Seeing the wood for the trees: a forest of methods for optimization and omic-network integration in metabolic modelling. *Briefings in bioinformatics*, 19(6):1218–1235, 2017.

[8] Supreeta Vijayakumar, Max Conway, Pietro Lió, and Claudio Angione. Optimization of multi-omic genome-scale models: methodologies, hands-on tutorial, and perspectives. In *Metabolic Network Reconstruction and Modeling*, pages 389–408. Humana Press, New York, NY, 2018.

[9] Olusegun Oshota, Max Conway, Maria Fookes, Fernanda Schreiber, Roy R Chaudhuri, Lu Yu, Fiona JE Morgan, Simon Clare, Jyoti Choudhary, Nicholas R Thomson, et al. Transcriptome and proteome analysis of salmonella enterica serovar typhimurium systemic infection of wild type and immune-deficient mice. *PloS one*, 12(8):e0181365, 2017.

[10] Max Conway, Claudio Angione, and Pietro Liò. Iterative multi level calibration of metabolic networks. *Current Bioinformatics*, 11(1):93–105, 2016.

[11] Alessio Mancini, Filmon Eyassu, Maxwell Conway, Annalisa Occhipinti, Pietro Liò, Claudio Angione, and Sandra Pucciarelli. Ciliategem: an open-project and a tool for predictions of ciliate metabolic variations and experimental condition design. *BMC bioinformatics*, 19(15):442, 2018.

[12] Claudio Angione, Max Conway, and Pietro Lió. Multiplex methods provide effective integration of multi-omic data in genome-scale models. *BMC bioinformatics*, 17(4):83, 2016.

[13] Claudio Angione, Pietro Liò, Sandra Pucciarelli, Basarbatu Can, Maxwell Conway, Marina Lotti, Habib Bokhari, Alessio Mancini, Ugur Sezerman, and Andrea Telatin. Bioinformatics challenges and potentialities in studying extreme environments. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 205–219. Springer, 2015.

[14] Jole Costanza, Giovanni Carapezza, Claudio Angione, Pietro Lió, and Giuseppe Nicosia. Robust design of microbial strains. *Bioinformatics*, 28(23):3097–3104, 2012.

[15] Igor Saggese, Elisa Bona, Max Conway, Francesco Favero, Marco Ladetto, Pietro Liò, Giovanni Manzini, and Flavio Mignone. Stable: a novel approach to de novo assembly of rna-seq data and its application in a metabolic model network based metatranscriptomic workflow. *BMC bioinformatics*, 19(7):184, 2018.

[16] Gustav Kirchhoff. Liv. on the motion of electricity in wires. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 13(88):393–412, 1857.

[17] Nathan E Lewis, Harish Nagarajan, and Bernhard O Palsson. Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods. *Nature Reviews Microbiology*, 10(4):291, 2012.

[18] Jürgen Zanghellini, David E Ruckerbauer, Michael Hanscho, and Christian Jungreuthmayer. Elementary flux modes in a nutshell: properties, calculation and applications. *Biotechnology journal*, 8(9):1009–1016, 2013.

[19] Andrew Makhorin. Glpk (gnu linear programming kit), 2000. *B B*, 2014.

[20] Stefan Theussl and Kurt Hornik. *Rglpk: R/GNU Linear Programming Kit Interface*, 2017. R package version 0.6-3.

[21] Stefan Theussl. *ROI.plugin.glpk: 'ROI' Plug-in 'GLPK'*, 2017. R package version 0.3-0.

[22] Anqi Fu and Balasubramanian Narasimhan. *ECOSolveR: Embedded Conic Solver in R*, 2018. R package version 0.4.

[23] Florian Schwendinger. *ROI.plugin.ecos: 'ECOS' Plugin for the 'R' Optimization Infrastructure*, 2017. R package version 0.3-0.

[24] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.

[25] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[26] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[27] Jan Schellenberger, Junyoung O Park, Tom M Conrad, and Bernhard Ø Palsson. Bigg: a biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions. *BMC bioinformatics*, 11(1):213, 2010.

[28] Peter D Karp, Christos A Ouzounis, Caroline Moore-Kochlacs, Leon Goldovsky, Pallavi Kaipa, Dag Ahrén, Sophia Tsoka, Nikos Darzentas, Victor Kunin, and Núria López-Bigas. Expansion of the biocyc collection of pathway/genome databases to 160 genomes. *Nucleic acids research*, 33(19):6083–6089, 2005.

[29] Sébastien Moretti, Olivier Martin, T Van Du Tran, Alan Bridge, Anne Morgat, and Marco Pagni. Metanetx/mnxref–reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks. *Nucleic acids research*, 44(D1):D523–D526, 2015.

[30] Christopher S Henry, Matthew DeJongh, Aaron A Best, Paul M Frybarger, Ben Linsay, and Rick L Stevens. High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nature biotechnology*, 28(9):977, 2010.

[31] Sylvain Prigent, Clémence Frioux, Simon M Dittami, Sven Thiele, Abdelhalim Larhlimi, Guillaume Collet, Fabien Gutknecht, Jeanne Got, Damien Eveillard,

Jérémie Bourdon, et al. Meneco, a topology-based gap-filling tool applicable to degraded genome-wide metabolic networks. *PLoS computational biology*, 13(1):e1005276, 2017.

[32] Bashir Sajo Mienda. Genome-scale metabolic models as platforms for strain design and biological discovery. *Journal of Biomolecular Structure and Dynamics*, 35(9):1863–1873, 2017.

[33] Vinay Satish Kumar and Costas D Maranas. Growmatch: an automated method for reconciling in silico/in vivo growth predictions. *PLoS computational biology*, 5(3):e1000308, 2009.

[34] Jennifer L Reed, Trina R Patel, Keri H Chen, Andrew R Joyce, Margaret K Applebee, Christopher D Herring, Olivia T Bui, Eric M Knight, Stephen S Fong, and Bernhard O Palsson. Systems approach to refining genome annotation. *Proceedings of the National Academy of Sciences*, 103(46):17480–17484, 2006.

[35] Markus J Herrgård, Stephen S Fong, and Bernhard Ø Palsson. Identification of genome-scale metabolic network models using experimentally measured flux profiles. *PLoS computational biology*, 2(7):e72, 2006.

[36] Edward J OBrien, Jonathan M Monk, and Bernhard O Palsson. Using genome-scale models to predict biological capabilities. *Cell*, 161(5):971–987, 2015.

[37] Vinay Satish Kumar, Madhukar S Dasika, and Costas D Maranas. Optimization based automated curation of metabolic reconstructions. *BMC bioinformatics*, 8(1):212, 2007.

[38] Ines Thiele, Nikos Vlassis, and Ronan MT Fleming. fastgapfill: efficient gap filling in metabolic networks. *Bioinformatics*, 30(17):2529–2531, 2014.

[39] Edward Vitkin and Tomer Shlomi. Mirage: a functional genomics-based approach for metabolic network model reconstruction and its application to cyanobacteria networks. *Genome biology*, 13(11):R111, 2012.

[40] Jason A Papin, Joerg Stelling, Nathan D Price, Steffen Klamt, Stefan Schuster, and Bernhard O Palsson. Comparison of network-based pathway analysis methods. *Trends in biotechnology*, 22(8):400–405, 2004.

[41] Zhaoyuan Fang, Weidong Tian, and Hongbin Ji. A network-based gene-weighting approach for pathway analysis. *Cell research*, 22(3):565, 2012.

[42] Qiaosheng Zhang, Jie Li, Haozhe Xie, Hanqing Xue, and Yadong Wang. A network-based pathway-expanding approach for pathway analysis. *BMC bioinformatics*, 17(17):536, 2016.

[43] Lu Liu and Jianhua Ruan. Network-based pathway enrichment analysis. In *2013 IEEE International Conference on Bioinformatics and Biomedicine*, pages 218–221. IEEE, 2013.

[44] Jing Ma, Ali Shojaie, and George Michailidis. Network-based pathway enrichment analysis with incomplete network information. *Bioinformatics*, 32(20):3165–3174, 2016.

[45] Sabine Pérès, François Vallée, Marie Beurton-Aimar, and Jean-Pierre Mazat. Acom: a classification method for elementary flux modes based on motif finding. *Biosystems*, 103(3):410–419, 2011.

[46] Christoph Kaleta, Luis Filipe De Figueiredo, Jörn Behre, and Stefan Schuster. Efmevolver: Computing elementary flux modes in genome-scale metabolic networks. In *Lecture Notes in Informatics-Proceedings*, volume 157, pages 179–189, 2009.

[47] Luis F De Figueiredo, Adam Podhorski, Angel Rubio, Christoph Kaleta, John E Beasley, Stefan Schuster, and Francisco J Planes. Computing the shortest elementary flux modes in genome-scale metabolic networks. *Bioinformatics*, 25(23):3158–3165, 2009.

[48] Clemens Wagner and Robert Urbanczik. The geometry of the flux cone of a metabolic network. *Biophysical journal*, 89(6):3837–3845, 2005.

[49] Alberto Rezola, Luis F de Figueiredo, M Brock, Jon Pey, Adam Podhorski, Christoph Wittmann, Stefan Schuster, Alexander Bockmayr, and Francisco J Planes. Exploring metabolic pathways in genome-scale networks via generating flux modes. *Bioinformatics*, 27(4):534–540, 2010.

[50] Abdelhalim Larhlimi and Alexander Bockmayr. A new constraint-based description of the steady-state flux cone of metabolic networks. *Discrete Applied Mathematics*, 157(10):2257–2266, 2009.

[51] Robert Urbanczik and Clemens Wagner. An improved algorithm for stoichiometric network analysis: theory and applications. *Bioinformatics*, 21(7):1203–1210, 2004.

[52] José Francisco Hidalgo, Francisco Guil, and José Manuel García. A new approach to obtaining efms using graph methods based on the shortest path between end nodes. *Genomics and Computational Biology*, 2(1):e30–e30, 2016.

[53] Alberto Rezola, Jon Pey, Luis Tobalina, Ángel Rubio, John E Beasley, and Francisco J Planes. Advances in network-based metabolic pathway analysis and gene expression data integration. *Briefings in bioinformatics*, 16(2):265–279, 2014.

[54] Matthias P Gerstl, David E Ruckerbauer, Diethard Mattanovich, Christian Jungreuthmayer, and Jürgen Zanghellini. Metabolomics integrated elementary flux mode analysis in large metabolic networks. *Scientific reports*, 5:8930, 2015.

[55] Anne Kümmel, Sven Panke, and Matthias Heinemann. Putative regulatory sites unraveled by network-embedded thermodynamic analysis of metabolome data. *Molecular systems biology*, 2(1), 2006.

[56] Matthias P Gerstl, Christian Jungreuthmayer, Stefan Müller, and Jürgen Zanghellini. Which sets of elementary flux modes form thermodynamically feasible flux distributions? *The FEBS journal*, 283(9):1782–1794, 2016.

[57] Christophe H Schilling, David Letscher, and Bernhard Ø Palsson. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of theoretical biology*, 203(3):229–248, 2000.

[58] Bruce L Clarke. Stoichiometric network analysis. *Cell biophysics*, 12(1):237–253, 1988.

[59] Sangaalofa T Clark and Wynand S Verwoerd. Minimal cut sets and the use of failure modes in metabolic networks. *Metabolites*, 2(3):567–595, 2012.

[60] Christoph Kaleta, Luís Filipe de Figueiredo, and Stefan Schuster. Can the whole be less than the sum of its parts? pathway analysis in genome-scale metabolic networks using elementary flux patterns. *Genome research*, 19(10):1872–1883, 2009.

[61] Marnix H Medema, Renske Van Raaphorst, Eriko Takano, and Rainer Breitling. Computational tools for the synthetic design of biochemical pathways. *Nature Reviews Microbiology*, 10(3):191, 2012.

[62] Miguel A Campodonico, Barbara A Andrews, Juan A Asenjo, Bernhard O Palsson, and Adam M Feist. Generation of an atlas for commodity chemical production in escherichia coli and a novel pathway prediction algorithm, gem-path. *Metabolic engineering*, 25:140–158, 2014.

[63] Vassily Hatzimanikatis, Chunhui Li, Justin A Ionita, Christopher S Henry, Matthew D Jankowski, and Linda J Broadbelt. Exploring the diversity of complex metabolic networks. *Bioinformatics*, 21(8):1603–1609, 2005.

[64] Alfredo Braunstein, Roberto Mulet, and Andrea Pagnani. Estimating the size of the solution space of metabolic networks. *BMC bioinformatics*, 9(1):240, 2008.

[65] David W Schryer, Marko Vendelin, and Pearu Peterson. Symbolic flux analysis for genome-scale metabolic networks. *BMC systems biology*, 5(1):81, 2011.

[66] Areejit Samal, João F Matias Rodrigues, Jürgen Jost, Olivier C Martin, and Andreas Wagner. Genotype networks in metabolic reaction spaces. *BMC systems biology*, 4(1):30, 2010.

[67] Daniele De Martino, Matteo Mori, and Valerio Parisi. Uniform sampling of steady states in metabolic networks: heterogeneous scales and rounding. *PloS one*, 10(4):e0122670, 2015.

[68] Maike K Aurich, Ronan MT Fleming, and Ines Thiele. Metabotools: a comprehensive toolbox for analysis of genome-scale metabolic models. *Frontiers in physiology*, 7:327, 2016.

[69] Monica L Mo, Bernhard Ø Palsson, and Markus J Herrgård. Connecting extracellular metabolomic measurements to intracellular flux states in yeast. *BMC systems biology*, 3(1):37, 2009.

[70] Maike K Aurich, Giuseppe Paglia, Óttar Rolfsson, Sigrún Hrafnsdóttir, Manuela Magnúsdóttir, Magdalena M Stefaniak, Bernhard Ø Palsson, Ronan MT Fleming, and Ines Thiele. Prediction of intracellular metabolic states from extracellular metabolomic data. *Metabolomics*, 11(3):603–619, 2015.

[71] Łukasz P Zieliński, Anthony C Smith, Alexander G Smith, and Alan J Robinson. Metabolic flexibility of mitochondrial respiratory chain disorders predicted by computer modelling. *Mitochondrion*, 31:45–55, 2016.

[72] Radhakrishnan Mahadevan, Jeremy S Edwards, and Francis J Doyle III. Dynamic flux balance analysis of diauxic growth in escherichia coli. *Biophysical journal*, 83(3):1331–1340, 2002.

[73] Wolfgang Wiechert. 13c metabolic flux analysis. *Metabolic engineering*, 3(3):195–206, 2001.

[74] Kai Zhuang, Mounir Izallalen, Paula Mouser, Hanno Richter, Carla Risso, Radhakrishnan Mahadevan, and Derek R Lovley. Genome-scale dynamic modeling of the competition between rhodoferax and geobacter in anoxic subsurface environments. *The ISME journal*, 5(2):305, 2011.

[75] Jennifer L Reed. Shrinking the metabolic solution space using experimental datasets. *PLoS computational biology*, 8(8):e1002662, 2012.

[76] Anthony P Burgard, Shankar Vaidyaraman, and Costas D Maranas. Minimal reaction sets for escherichia coli metabolism under different growth requirements and uptake environments. *Biotechnology progress*, 17(5):791–797, 2001.

[77] R Mahadevan and CH Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic engineering*, 5(4):264–276, 2003.

[78] Arne C Müller and Alexander Bockmayr. Fast thermodynamically constrained flux variability analysis. *Bioinformatics*, 29(7):903–909, 2013.

[79] Christopher S Henry, Linda J Broadbelt, and Vassily Hatzimanikatis. Thermodynamics-based metabolic flux analysis. *Biophysical journal*, 92(5):1792–1805, 2007.

[80] Meric Ataman and Vassily Hatzimanikatis. Heading in the right direction: thermodynamics-based network analysis and pathway engineering. *Current opinion in biotechnology*, 36:176–182, 2015.

[81] Nathan D Price, Iman Famili, Daniel A Beard, and Bernhard Ø Palsson. Extreme pathways and kirchhoff's second law. *Biophysical journal*, 83(5):2879–2882, 2002.

[82] Jan Schellenberger, Nathan E Lewis, and Bernhard Ø Palsson. Elimination of thermodynamically infeasible loops in steady-state metabolic models. *Biophysical journal*, 100(3):544–553, 2011.

[83] Daniel A Beard, Shou-dan Liang, and Hong Qian. Energy balance for analysis of complex metabolic networks. *Biophysical journal*, 83(1):79–86, 2002.

[84] Nathan E Lewis, Kim K Hixson, Tom M Conrad, Joshua A Lerman, Pep Charusanti, Ashoka D Polpitiya, Joshua N Adkins, Gunnar Schramm, Samuel O Purvine, Daniel Lopez-Ferrer, et al. Omic data from evolved e. coli are consistent with computed optimal growth from genome-scale models. *Molecular systems biology*, 6(1):390, 2010.

[85] Marco Rügen, Alexander Bockmayr, and Ralf Steuer. Elucidating temporal resource allocation and diurnal dynamics in phototrophic metabolism using conditional fba. *Scientific reports*, 5:15247, 2015.

[86] Alexandra-M Reimers, Henning Knoop, Alexander Bockmayr, and Ralf Steuer. Evaluating the stoichiometric and energetic constraints of cyanobacterial diurnal growth. *arXiv preprint arXiv:1610.06859*, 2016.

[87] Anne Goelzer and Vincent Fromion. Bacterial growth rate reflects a bottleneck in resource allocation. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1810(10):978–988, 2011.

[88] Matteo Mori, Terence Hwa, Olivier C Martin, Andrea De Martino, and Enzo Marinari. Constrained allocation flux balance analysis. *PLoS computational biology*, 12(6):e1004913, 2016.

[89] Jenni Heino, Knarik Tunyan, Daniela Calvetti, and Erkki Somersalo. Bayesian flux balance analysis applied to a skeletal muscle metabolic model. *Journal of theoretical biology*, 248(1):91–110, 2007.

[90] Jenni Heino, Daniela Calvetti, and Erkki Somersalo. Metabolica: a statistical research tool for analyzing metabolic networks. *Computer methods and programs in biomedicine*, 97(2):151–167, 2010.

[91] Elsa W Birch, Madeleine Udell, and Markus W Covert. Incorporation of flexible objectives and time-linked simulation with flux balance analysis. *Journal of theoretical biology*, 345:12–21, 2014.

[92] Tomer Shlomi, Yariv Eisenberg, Roded Sharan, and Eytan Ruppin. A genome-scale computational study of the interplay between transcriptional regulation and metabolism. *Molecular systems biology*, 3(1):101, 2007.

[93] Markus W Covert, Nan Xiao, Tiffany J Chen, and Jonathan R Karr. Integrating metabolic, transcriptional regulatory and signal transduction models in escherichia coli. *Bioinformatics*, 24(18):2044–2050, 2008.

[94] Jong Min Lee, Erwin P Gianchandani, James A Eddy, and Jason A Papin. Dynamic analysis of integrated signaling, metabolic, and regulatory networks. *PLoS computational biology*, 4(5):e1000086, 2008.

[95] Sriram Chandrasekaran and Nathan D Price. Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in escherichia coli and mycobacterium tuberculosis. *Proceedings of the National Academy of Sciences*, 107(41):17845–17850, 2010.

[96] Sriram Chandrasekaran and Nathan D Price. Metabolic constraint-based refinement of transcriptional regulatory networks. *PLoS computational biology*, 9(12):e1003370, 2013.

[97] Ciaran P Fisher, Nicholas J Plant, J Bernadette Moore, and Andrzej M Kierzek. Qsspn: dynamic simulation of molecular interaction networks describing gene regulation, signalling and whole-cell metabolism in human cells. *Bioinformatics*, 29(24):3181–3190, 2013.

[98] Lucas Marmiesse, Rémi Peyraud, and Ludovic Cottret. Flexflux: combining metabolic flux and regulatory network analyses. *BMC systems biology*, 9(1):93, 2015.

[99] Huihai Wu, Axel Von Kamp, Vytautas Leoncikas, Wataru Mori, Nilgun Sahin, Albert Gevorgyan, Catherine Linley, Marek Grabowski, Ahmad A Mannan, Nicholas Stoy, et al. Mufins: multi-formalism interaction network simulator. *NPJ systems biology and applications*, 2:16032, 2016.

[100] Rogier JP van Berlo, Dick de Ridder, Jean-Marc Daran, Pascale AS Daran-Lapujade, Bas Teusink, and Marcel JT Reinders. Predicting metabolic fluxes using gene expression differences as constraints. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(1):206–216, 2011.

[101] Ehsan Motamedian, Maryam Mohammadi, Seyed Abbas Shojaosadati, and Mona Heydari. Trfba: an algorithm to integrate genome-scale metabolic and transcriptional regulatory networks with incorporation of expression data. *Bioinformatics*, 33(7):1057–1063, 2016.

[102] Robert Petryszak, Maria Keays, Y Amy Tang, Nuno A Fonseca, Elisabet Barrera, Tony Burdett, Anja Füllgrabe, Alfonso Muñoz-Pomer Fuentes, Simon Jupp, Satu Koskinen, et al. Expression atlas updatean integrated database of gene and protein expression in humans, animals and plants. *Nucleic acids research*, 44(D1):D746–D752, 2015.

[103] Nikolay Kolesnikov, Emma Hastings, Maria Keays, Olga Melnichuk, Y Amy Tang, Eleanor Williams, Miroslaw Dylag, Natalja Kurbatova, Marco Brandizi, Tony Burdett, et al. Arrayexpress updatesimplifying data submissions. *Nucleic acids research*, 43(D1):D1113–D1116, 2014.

[104] Tanya Barrett, Stephen E Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F Kim, Maxim Tomashevsky, Kimberly A Marshall, Katherine H Phillippy, Patti M Sherman, Michelle Holko, et al. Ncbi geo: archive for functional genomics data setsupdate. *Nucleic acids research*, 41(D1):D991–D995, 2012.

[105] Rong Chen, Rohan Mallelwar, Ajit Thosar, Shivkumar Venkatasubrahmanyam, and Atul J Butte. Genechaser: identifying all biological and clinical conditions in which genes of interest are differentially expressed. *BMC bioinformatics*, 9(1):548, 2008.

[106] Jesse M Engreitz, Rong Chen, Alexander A Morgan, Joel T Dudley, Rohan Mallelwar, and Atul J Butte. Profilechaser: searching microarray repositories based on genome-wide patterns of differential expression. *Bioinformatics*, 27(23):3317–3318, 2011.

[107] Johan Rung and Alvis Brazma. Reuse of public genome-wide gene expression data. *Nature Reviews Genetics*, 14(2):89, 2013.

[108] Ali Salehzadeh-Yazdi, Yazdan Asgari, Ali Akbar Saboury, and Ali Masoudi-Nejad. Computational analysis of reciprocal association of metabolism and epigenetics in the budding yeast: a genome-scale metabolic model (gsmm) approach. *PloS one*, 9(11):e111686, 2014.

[109] RP Vivek-Ananth and Areejit Samal. Advances in the integration of transcriptional regulatory information into genome-scale metabolic models. *Biosystems*, 147:1–10, 2016.

[110] Min Kyung Kim and Desmond S Lun. Methods for integration of transcriptomic data in genome-scale metabolic models. *Computational and structural biotechnology journal*, 11(18):59–65, 2014.

[111] Scott A Becker and Bernhard O Palsson. Context-specific metabolic networks are consistent with experiments. *PLoS computational biology*, 4(5):e1000082, 2008.

[112] Brian J Schmidt, Ali Ebrahim, Thomas O Metz, Joshua N Adkins, Bernhard Ø Palsson, and Daniel R Hyduke. Gim3e: condition-specific models of cellular metabolism developed from metabolomics and expression data. *Bioinformatics*, 29(22):2900–2908, 2013.

[113] Tomer Shlomi, Moran N Cabili, Markus J Herrgård, Bernhard Ø Palsson, and Eytan Ruppin. Network-based prediction of human tissue-specific metabolism. *Nature biotechnology*, 26(9):1003, 2008.

[114] Anna S Blazier and Jason A Papin. Integration of expression data in genome-scale metabolic network reconstructions. *Frontiers in physiology*, 3:299, 2012.

[115] Hadas Zur, Eytan Ruppin, and Tomer Shlomi. imat: an integrative metabolic analysis tool. *Bioinformatics*, 26(24):3140–3142, 2010.

[116] Rasmus Agren, Sergio Bordel, Adil Mardinoglu, Natapol Pornputtapong, Intawat Nookaew, and Jens Nielsen. Reconstruction of genome-scale active metabolic networks for 69 human cell types and 16 cancer types using init. *PLoS computational biology*, 8(5):e1002518, 2012.

[117] Daniel Machado and Markus Herrgård. Systematic evaluation of methods for integration of transcriptomic data into constraint-based models of metabolism. *PLoS computational biology*, 10(4):e1003580, 2014.

[118] Caroline Colijn, Aaron Brandes, Jeremy Zucker, Desmond S Lun, Brian Weiner, Maha R Farhat, Tan-Yun Cheng, D Branch Moody, Megan Murray, and James E Galagan. Interpreting expression data with metabolic flux models: predicting mycobacterium tuberculosis mycolic acid production. *PLoS computational biology*, 5(8):e1000489, 2009.

[119] Min Kyung Kim, Anatoliy Lane, James J Kelley, and Desmond S Lun. E-flux2 and spot: validated methods for inferring intracellular metabolic flux distributions from transcriptomic data. *PloS one*, 11(6):e0157101, 2016.

[120] Claudio Angione and Pietro Lió. Predictive analytics of environmental adaptability in multi-omic network models. *Scientific reports*, 5:15147, 2015.

[121] Brandon E Barker, Narayanan Sadagopan, Yiping Wang, Kieran Smallbone, Christopher R Myers, Hongwei Xi, Jason W Locasale, and Zhenglong Gu. A robust and efficient method for estimating enzyme complex abundance and metabolic flux from expression data. *Computational biology and chemistry*, 59:98–112, 2015.

[122] Dave Lee, Kieran Smallbone, Warwick B Dunn, Ettore Murabito, Catherine L Winder, Douglas B Kell, Pedro Mendes, and Neil Swainston. Improving metabolic flux predictions using absolute gene expression data. *BMC systems biology*, 6(1):73, 2012.

[123] Keren Yizhak, Edoardo Gaude, Sylvia Le Dévédec, Yedael Y Waldman, Gideon Y Stein, Bob van de Water, Christian Frezza, and Eytan Ruppin. Phenotype-based cell-specific metabolic modeling reveals metabolic liabilities of cancer. *Elife*, 3:e03641, 2014.

[124] Nikos Vlassis, Maria Pires Pacheco, and Thomas Sauter. Fast reconstruction of compact context-specific metabolic network models. *PLoS computational biology*, 10(1):e1003424, 2014.

[125] Maria Pires Pacheco, Elisabeth John, Tony Kaoma, Merja Heinäniemi, Nathalie Nicot, Laurent Vallar, Jean-Luc Bueb, Lasse Sinkkonen, and Thomas Sauter. Integrated metabolic modelling reveals cell-type specific epigenetic control points of the macrophage metabolic network. *BMC genomics*, 16(1):809, 2015.

[126] Livnat Jerby, Tomer Shlomi, and Eytan Ruppin. Computational reconstruction of tissue-specific metabolic models: application to human liver metabolism. *Molecular systems biology*, 6(1):401, 2010.

[127] Yuliang Wang, James A Eddy, and Nathan D Price. Reconstruction of genome-scale metabolic models for 126 human tissues using mcadre. *BMC systems biology*, 6(1):153, 2012.

[128] Jonathan M Dreyfuss, Jeremy D Zucker, Heather M Hood, Linda R Ocasio, Matthew S Sachs, and James E Galagan. Reconstruction and validation of a genome-scale metabolic model for the filamentous fungus neurospora crassa using farm. *PLoS computational biology*, 9(7):e1003126, 2013.

[129] André Schultz and Amina A Qutub. Reconstruction of tissue-specific metabolic networks using corda. *PLoS computational biology*, 12(3):e1004808, 2016.

[130] Semidán Robaina Estévez and Zoran Nikoloski. Context-specific metabolic model extraction based on regularized least squares optimization. *PloS one*, 10(7):e0131875, 2015.

[131] Weihua Guo and Xueyang Feng. Om-fba: integrate transcriptomics data with flux balance analysis to decipher the cell metabolism. *PloS one*, 11(4):e0154188, 2016.

[132] Sebastian MB Nijman. Synthetic lethality: general principles, utility and detection using genetic screens in human cells. *FEBS letters*, 585(1):1–6, 2011.

[133] Patrick F Suthers, Alireza Zomorrodi, and Costas D Maranas. Genome-scale gene/reaction essentiality and synthetic lethality analysis. *Molecular systems biology*, 5(1):301, 2009.

[134] Aditya Pratapa, Shankar Balachandran, and Karthik Raman. Fast-sl: an efficient algorithm to identify synthetic lethal sets in metabolic networks. *Bioinformatics*, 31(20):3299–3305, 2015.

[135] Luis Tobalina, Jon Pey, and Francisco J Planes. Direct calculation of minimal cut sets involving a specific reaction knock-out. *Bioinformatics*, 32(13):2001–2007, 2016.

[136] Livnat Jerby-Arnon, Nadja Pfetzer, Yedael Y Waldman, Lynn McGarry, Daniel James, Emma Shanks, Brinton Seashore-Ludlow, Adam Weinstock, Tamar Geiger, Paul A Clemons, et al. Predicting cancer-specific vulnerability via data-driven detection of synthetic lethality. *Cell*, 158(5):1199–1209, 2014.

[137] Wout Megchelenbrink, Rotem Katzir, Xiaowen Lu, Eytan Ruppin, and Richard A Notebaart. Synthetic dosage lethality in the human metabolic network is highly predictive of tumor growth and cancer patient survival. *Proceedings of the National Academy of Sciences*, 112(39):12217–12222, 2015.

[138] Alpan Raval and Animesh Ray. *Introduction to biological networks*. Chapman and Hall/CRC, 2016.

[139] Daniel Segre, Dennis Vitkup, and George M Church. Analysis of optimality in natural and perturbed metabolic networks. *Proceedings of the National Academy of Sciences*, 99(23):15112–15117, 2002.

[140] Keren Yizhak, Tomer Benyamini, Wolfram Liebermeister, Eytan Ruppin, and Tomer Shlomi. Integrating quantitative proteomics and metabolomics with a genome-scale metabolic network model. *Bioinformatics*, 26(12):i255–i260, 2010.

[141] Tomer Shlomi, Omer Berkman, and Eytan Ruppin. Regulatory on/off minimization of metabolic flux changes after genetic perturbations. *Proceedings of the National Academy of Sciences*, 102(21):7695–7700, 2005.

[142] Michael J McAnulty, Jiun Y Yen, Benjamin G Freedman, and Ryan S Senger. Genome-scale modeling using flux ratio constraints to enable metabolic engineering of clostridial metabolism in silico. *BMC systems biology*, 6(1):42, 2012.

[143] Jiun Y Yen, Hadi Nazem-Bokaee, Benjamin G Freedman, Ahmad IM Athamneh, and Ryan S Senger. Deriving metabolic engineering strategies from genome-scale modeling with flux ratio constraints. *Biotechnology journal*, 8(5):581–594, 2013.

[144] Joonhoon Kim and Jennifer L Reed. Relatch: relative optimality in metabolic networks explains robust metabolic and regulatory responses to perturbations. *Genome biology*, 13(9):R78, 2012.

[145] Reza Miraskarshahi, Hooman Zabeti, Tamon Stephen, and Leonid Chindelevitch. Mcsˆ 2: Minimal coordinated supports for fast enumeration of minimal cut sets in metabolic networks. *BioRxiv*, page 471250, 2019.

[146] Alfredo Braunstein, Anna Paola Muntoni, and Andrea Pagnani. An analytic approximation of the feasible space of metabolic networks. *Nature communications*, 8:14915, 2017.

[147] Jennifer L Reed and Bernhard Ø Palsson. Genome-scale in silico models of e. coli have multiple equivalent phenotypic states: assessment of correlated reaction subsets that comprise network states. *Genome research*, 14(9):1797–1805, 2004.

[148] Jan Schellenberger and Bernhard Ø Palsson. Use of randomized sampling for analysis of metabolic networks. *Journal of biological chemistry*, 284(9):5457–5461, 2009.

[149] Sharon J Wiback, Iman Famili, Harvey J Greenberg, and Bernhard Ø Palsson. Monte carlo sampling can be used to determine the size and shape of the steady-state flux space. *Journal of theoretical biology*, 228(4):437–447, 2004.

[150] Jeremy S Edwards, Rafael U Ibarra, and Bernhard O Palsson. In silico predictions of escherichia coli metabolic capabilities are consistent with experimental data. *Nature biotechnology*, 19(2):125, 2001.

[151] Kristof Engelen, Qiang Fu, Pieter Meysman, Aminael Sánchez-Rodríguez, Riet De Smet, Karen Lemmens, Ana Carolina Fierro, and Kathleen Marchal. Colombos: access port for cross-platform bacterial expression compendia. *PLoS One*, 6(7):e20938, 2011.

[152] Christine Vogel and Edward M Marcotte. Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nature reviews genetics*, 13(4):227, 2012.

[153] James D Evans. *Straightforward statistics for the behavioral sciences*. Thomson Brooks/Cole Publishing Co, 1996.

[154] Michael Lynch, Matthew S Ackerman, Jean-Francois Gout, Hongan Long, Way Sung, W Kelley Thomas, and Patricia L Foster. Genetic drift, selection and the evolution of the mutation rate. *Nature Reviews Genetics*, 17(11):704–714, 2016.

[155] Max Conway. *fbar: An Extensible Approach to Flux Balance Analysis*, 2017. http://maxconway.github.io/fbar/, https://github.com/maxconway/fbar.

[156] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards. *Artificial intelligence: a modern approach*, volume 2. Prentice hall Upper Saddle River, 2003.

[157] Rafael U Ibarra, Jeremy S Edwards, and Bernhard O Palsson. Escherichia coli k-12 undergoes adaptive evolution to achieve in silico predicted optimal growth. *Nature*, 420(6912):186, 2002.

[158] Jeffrey D Orth, Tom M Conrad, Jessica Na, Joshua A Lerman, Hojung Nam, Adam M Feist, and Bernhard Ø Palsson. A comprehensive genome-scale reconstruction of escherichia coli metabolism2011. *Molecular systems biology*, 7(1):535, 2011.

[159] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[160] Gérard Biau, Aurélie Fischer, Benjamin Guedj, and James Malley. Cobra: a nonlinear aggregation strategy. *arXiv preprint arXiv:1303.2236*, 2013.

[161] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *International symposium on computer and information sciences*, pages 284–293. Springer, 2005.

[162] Bo Wang, Aziz Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. *SNFtool: Similarity Network Fusion*, 2017. R package version 2.2.1.

[163] James E Ferrell Jr. Feedback regulation of opposing enzymes generates robust, all-or-none bistable responses. *Current Biology*, 18(6):R244–R245, 2008.

[164] Terry Therneau and Beth Atkinson. *rpart: Recursive Partitioning and Regression Trees*, 2018. R package version 4.1-13.

[165] Sivaraman Balakrishnan, Min Xu, Akshay Krishnamurthy, and Aarti Singh. Noise thresholds for spectral clustering. In *Advances in Neural Information Processing Systems*, pages 954–962, 2011.

[166] Yoli Shavit, Barnabas James Walker, and Pietro Lio. Hierarchical block matrices as efficient representations of chromosome topologies and their application for 3c data integration. *Bioinformatics*, 32(8):1121–1129, 2015.

[167] Yali Wan. *Topics in Graph Clustering*. PhD thesis, 2017.

[168] Yali Wan and Marina Meila. Benchmarking recovery theorems for the dc-sbm. In *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2015.

[169] Qi Mao, Wei Zheng, Li Wang, Yunpeng Cai, Volker Mai, and Yijun Sun. Parallel hierarchical clustering in linearithmic time for large-scale sequence analysis. In *2015 IEEE International Conference on Data Mining*, pages 310–319. IEEE, 2015.

[170] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[171] George H John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier, 1994.

[172] Max Kuhn and Ross Quinlan. *Cubist: Rule- And Instance-Based Regression Modeling*, 2017. R package version 0.2.1.

[173] Jiang Su and Harry Zhang. A fast decision tree learning algorithm. In *AAAI*, volume 6, pages 500–505, 2006.

[174] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[175] Jing Zhao, Hong Yu, Jian-Hua Luo, Zhi-Wei Cao, and Yi-Xue Li. Hierarchical modularity of nested bow-ties in metabolic networks. *BMC bioinformatics*, 7(1):386, 2006.

[176] Yoli Shavit. *hbm: Hierarchical Block Matrix Analysis*, 2015. R package version 1.0.

[177] Kurt Hornik, David Meyer, Florian Schwendinger, and Stefan Theussl. *ROI: R Optimization Infrastructure*, 2018. R package version 0.3-1.

[178] Hadley Wickham. *tidyverse: Easily Install and Load the 'Tidyverse'*, 2017. R package version 1.2.1.

Corrections

In order to address the concerns in the report, I have expanded the thesis significantly. It is 60 % longer in terms of wordcount, twice as many equations and nearly three times as many citations. This document details what has been done to address each area of the report, however one of the most clear points from the report was that I needed to define and evaluate goals better. To address this, I have rewritten introductions and conclusions to try and make these goals clearer, and pushed the whole thesis towards more straightforward unsupervised learning evaluations. In particular:

- Section 5.7 validates the results of Chapter 5 to show that it has detected real clustering, and that this clustering is due to network structure itself, rather than genetic or regulator mechanisms. This appears to me to be biologically significant.
- Section 6.5 validates the results of Chapter 6 by comparing the clusters generated against manually labelled subsystems. My clustering method is found to outperform traditional methods.

1: Typos in abstract I have fixed the two typos in the abstract, and also largely rewritten it.

2: Lack of proper maths and equations I have added a lot more mathematical rigour. This includes:

- Equations 2.1 2.6 and 2.9 in page 30 in an extended definition of FBA in section 2.8.3.1
- Extended rigorous definitions of each sampling strategy in Chapter 3, based on the new definitions in the background chapter and in the new equation 3.6. This includes equation (3.9) and equations 3.12, 3.16 and 3.20 in section 3.2.6.
- Updates to the definitions of the case study sampling strategy in equation (3.27) and equation (3.30).
- New equations to improve the definition of the Local Network Learning approach in Chapter 6 including equations 6.1, 6.11, 6.12 and 6.15

3, 9:

- Lack of proper hypothesis/claims
- "In many cases I failed to understand what the goals were for the investigations"

I have added specific hypotheses to each chapter's introduction, and a discussion of these hypotheses in each conclusion. Chapter 4 has specific hypotheses and claims for each project. This was somewhat more difficult with Chapter 3, since the main test of the quality of its results is their usefulness in the other chapters, but I have attempted to make the goals more explicit.

4: Lack of definition of scientific question being addressed I have expanded the introductions and conclusions of each section, as well as to the overall chapter introduction and conclusion. The two main threads to this chapter are that useful information can be derived from sampling over metabolic networks, and that manual graphical approaches to interpreting large numbers of metabolic networks are not enough. Part of the reason for this chapter is to try and illustrate other approaches that I tried to this problem, since this is what motivated me to try and solve the problem of modelling the structure of ensembles of flow networks - there didn't appear to be any tools that solved this problem effectively.

5: Needs to specify specific shortfalls of previous tools I have added a substantially enhanced literature review in section 2.8 discussing what previous tools can and cannot do, and how the aims of this tool differ.

6: "The reader is instead referred to publications by the author for further detail. This should be included in the thesis" I have removed these statements, in sections 3.3.1 and 4.1, and brought in further information as required.

7: Need to state personal contribution in each paper I have done this, in section 1.2

8: The thesis cites only 61 papers I have added more citations, particularly in the expanded literature review in section 2.8. It now cites 178.

9: "In many cases I failed to understand what the goals were for the investigations" I have significantly expanded all introductions and conclusions with explicit goals and hypotheses to help to clarify this. I have also added validation sections (5.7 and 6.5) to provide more explicit goals.

10: "I didn't understand what we have learnt about network structures from the analyses carried out, neither biologically or computationally" In addition to lengthening the conclusions to each chapter, I have added sections 5.7 and 6.5, which validate the results in more concrete terms. Section 5.7 shows that the clusters detected in Chapter 5 are derived from related mechanisms in both experimental and simulated datasets, showing that we have learnt that they are inherent in the network structure. Section 6.5 validates the effectiveness of Chapter 6's clustering methodology by application to predicting known human labelled subsystems. This shows the effectiveness of predicting groupings of reactions based on real flux data, rather than purely on network structure.

11: "The main concern for both of us was 'so what" I have expanded the abstract, introduction and conclusion, as well as the relevant sections in each chapter to address this. In short, metabolic network structure and properties should be understood and evaluated in the context of the flows through them, and large scale sampling techniques can achieve this for both experimental and simulated data. The tools that I have developed allow structure to be found these large datasets, both in the structure of the network and in the resulting structure of the population.

12: "No attempt to show how Max wants his thesis to be judged and evaluated" I have extended all introductions and conclusions to try and make this clearer, and in particular added evaluation sections showing much more concrete evidence of techniques working.

13: "Suppose scientist A uses state of the art tools, and Scientist B uses Max's tools, how will scientist B be better off? They will be able to better identify structure in metabolic networks, both in how the reactions relate to each other (section 6.5), and by identifying population structure that is implied by the network (section 5.7). This will be possible across both experimental and simulated datasets, and they will be able to create simulate datasets much faster (section 3.4.1) than previously.

14: "did not adequately describe the goals and significance" I have expanded all introductions and conclusions to try and address this, with added focus on specific hypotheses.

15: "What is the scientific contribution here and how should it be evaluated" Although the aims of this research are somewhat novel, I have aimed to make the advantages of this more specific and concrete, such as speed comparisons (section 3.4.1), specific insights from clustering (section 5.7), and cluster evaluation via Adjusted Rand Index (section 6.5).

In terms of the overall scientific contribution, I feel that this is that statistical analysis of large sets of feasible metabolic network states should be a primary tool in the forefront of the toolbox for interpreting understanding metabolic networks, and I have tried to make this clear in the abstract, introduction and conclusion.

16: "Scientifically rigorous comparisons" The best example of this is the new section 6.5, which provides a detailed side by side comparison of my Hierarchical Block Matrix / Local Network Learning networks clustering with both a random baseline and other off the shelf clustering methods. Section 5.7 also adds a validation section to Chapter 5, but here it was more difficult to find an external comparison point, so the validation is primarily showing that the technique works.

17: "State what your hypotheses are" I have added introductions to each chapter (and section where appropriate) stating hypotheses and goals.

18: State what your methodology allows you to achieve that existing methods cannot I have now stated explicitly in the introduction the headline advantage: my methodology is agnostic to whether data is experimental or just a theoretical network, or indeed anything in between. This allows it to detect population structure that is inherent in the network (Chapter 5) or to condition network structure on simulated and experimental datasets (Chapter 6).

19: Remove vague statements like "This plot offers useful insight into network structure" I have removed all value judgements of this kind that I could find, or backed them up where that was possible.

20: What sort of researcher would be looking for what insights, for what purpose, and how do I show it's useful I have tried to be more specific about the real world implications of the results. For instance, in section 5.7, I identify specifically that the Fe^{2+} flux is a major correlate of the overall clustering found in both simulated and experimental datasets, but that its behaviour is different in each, and in section 6.5, I specifically test out the effectiveness of my clustering technique at labelling related reactions.

21: "The overall aim of this work was to demonstrate methodology that could be used to interpret multi-omic biological data" I have tried to be more focussed about my aims: statistical analysis of large datasets of metabolic networks has a significant advantage over purely network structure approaches because it can seamlessly incorporate both experimental and simulated data, and I aim to find techniques for this which provide

insights into real network structure. Sections 5.7 and 6.5 show justifications of why the network structure discovered is genuine.

22: Do not write e.g. "This is described in more detail in" I have removed every statement of this kind that I could find, and included the extra detail where it was needed.

23, 24. 25:

- There needs to be a much more systematic review of the state of the art of metabolic flux analysis, expanding the 5 lines on page 24 into several pages
- There should be a survey of key papers
- The work in the thesis should be put into the context of recent developments

I have added extensively to the background chapter with a review of related methods in section 2.8. Section 2.8.5 provides some more specific comparison with recent context.

26: The thesis proposes novel goals, so the literature review needs to go in depth into the goals *i.e.* the evaluation criteria for existing methods I have extended the literature review significantly, but I was unable to find any existing evaluation approach from the field seemed to fit what I was intending to do. For that reason I have instead approached evaluation using more general techniques for validating unsupervised learning.

In section 5.7 I did this in a relatively qualitative way by showing that the clustering is both real by showing graphically that the clustering is both real by showing the clear quantisation in the underlying data. I considered the use of statistical cluster quality measures such as the DaviesBouldin index or the Dunn index, but these require distance measures which are not well defined on this dataset, and I also considered repeatedly re-clustering, though this would have been time consuming and it was unclear at what stage in the process it would have been best to inject randomisation. In the end my conclusion was that the graphical approach was enough to show the data quantisation and that this data quantisation stemmed from similar mechanisms, which is something that I particularly wanted to show.

In section 6.5 I used the Adjusted Rand Index as an evaluation criteria, to measure the similarity of my clustering to human labelled subsystem labellings.

27: Make it clear what different approaches aim to achieve, their underlying assumptions, where they break down, and what gap you fill I have added a

larger literature review in section 2.8 to address this, in particular section 2.8.5 which reviews the most similar techniques.

28: Chapter 3: needs a formal mathematical statement of the basic optimization problem, listing variables and constants, and how they're being modified in each section I have now defined the basic optimization problem in Chapter 2 in equations 2.1 2.6 and 2.9. This is then reviewed in Chapter 3 in equation (3.6). I have then used these definitions as appropriate in the prose of Chapter 3 as well as in equations 3.9, 3.12, 3.16 and 3.20. Finally, I have also used these definitions to help describe the case studies in equation (3.27) and equation (3.30).

29: Chapter 3: needs a clear statement of the goal. It says 'In order to characterize natural phenotypes, we need to design a distribution from which to draw them. I have greatly expanded the introduction to Chapter 3 in order to explain the aims of this chapter: why these sampling techniques are needed, what properties they should have, and their merits relative to experimental data. Section 3.3 goes into further detail on the relative merits of experimental data sources, and section 3.4 adds more discussion on desirable properties for a distribution.

30: Chapter 3: do not use phrases like 'a more realistic solution' unless you can back it up I have justified or removed every statement of this kind that I could find. With respect to the specific claims in Chapter 3, I have attempted to soften my position from 'biologically realistic' to 'biologically justifiable'.

31: Chapter 3: only scenario that's clearly natural is where you start with natural distribution of fluxes, but isn't this already done? I have weakened my claims about my sampling strategies from 'natural' to 'biologically realistic', and at the same time described the weaknesses of experimental data in more detail to highlight the comparative advantages of my simulation strategies.

32: Chapter 3: Could make more modest claim 'I synthesise some variation, in order to usefully characterise the flow network I have moved in this direction, trying to make it clear that I want the variation to be biologically justifiable when compared the previous random uniform sampling, but I do not make any claims that this is the best method for this, since it would be extremely difficult to show that.

33: Chapter 3: section 3.3.3, note the lower bounds on ATP and overall growth in connection with possibility of no viable models I have explained this in more detail - most of the models that I used have at least some minimum flux bounds

above zero, or maximum flux bounds below zero, which means that it is possible to get infeasible solutions, since the all zeros solution is not viable. See section 3.3.3

34: Chapter 3: section 3.3.1, don't write down equation without first defining the terms I have dealt with this here in equation (3.25), and in all other equations.

35: Chapter 3: fig 3.2, density should have an asymptote rather than a peak For this figure, figure 3.2 in page 53, I moved from a kernel density plot to an empirical cumulative distribution plot. This required increasing the simulation size I used to generate it, but it means that the data is displayed exactly, rather than being smoothed.

36: Chapter 4: What is the aim and why? Evaluate the tools I have ensured that each section states explicit aims, and I have significantly expanded the conclusions to give a more detailed evaluation of the results.

37: Chapter 4: feels like a methods section, but missing an objective and evaluation I have been clearer in the introduction that this chapter is better thought of as three related projects, and lengthened both the section conclusions and the main chapter conclusion to make it clearer what was learned from them about what can and cannot be achieved with relatively small datasets and visual interpretation.

38: Chapter 4: very hard to understand how the operations in section 4.2.1 follow from the objective in the beginning of section 4.2 I have added more discussion in section 4.2.1 to make the goals of the normalisation clearer.

39: Chapter 5: What is the aim and why? How should it be judged I have significantly lengthened the introduction to provide a more specific hypothesis, making it clear that the goal of this chapter is to use SNF to detect population structure in metabolic networks, and to show that this structure is at least partly inherent in the network itself, rather than being just a sampling artefact. I have added a new section, section 5.7, to show that the network structure detected is not purely a clustering artefact, and show similarities between the experimental and simulated datasets which imply that the clustering structure is a property of the network.

40: Chapter 5: The heat maps in figures 5.2 and 5.3 are likely to give the appearance of clustering even when there is none I have added a new section, 5.7, which focusses on validating these figures. I created two new plots, figure 5.4a and

figure 5.4b, which display the same dataset, with no clustering, focussing on the relationship between biomass and Fe^{2+} exchange fluxes. This shows that some of the quantisation structure is still visible without the use of any clustering.

41: Chapter 5: Gives rise to clusters is a very crude way of evaluating whether two distributions are similar In the new validation section, 5.7, I have looked at the cause of the quantisation structure in the distributions, and focussed on a particular flux, Fe^{2+} exchange. This shows us something about how the underlying causes of the patterns in the distributions are similar, which provides stronger evidence that the similarities are not just a coincidence.

42: Chapter 5: If you want to make a claim about my method for synthesising variation mimics natural variation, discuss parameter tuning to fit the population I added a note about the normalisation for these two datasets in section 5.6. Essentially, the normalisation on experimental datasets is fairly aggressive, meaning that to tune the simulated dataset to match the experimental one, I need only replicate the normalisation so that they both end up in the same place. In this case the experimental dataset started out normalised to 0 mean, and 0 standard deviation, so I normalised the simulate dataset to the same, and then applied the same analysis from there on, transforming both to flux bounds using code based on equation (3.26)

43: Chapter 5: section 5.3.2-4: need to give a more precise setup I have expanded section 5.3.2 in order to give a more clear introduction to improve the readability and make definitions clearer.

44: Chapter 5: Maybe used weighted rather than biased for SNftool, because biased has already been used I have done this.

45: Chapter 5: section 5.5: explain use of spectral clustering I have added a comment in section 5.5 to say that I chose spectral clustering because it is recommended by Wang *et. al* [162].

46: Chapter 6: Aim and why I have extended the introduction to Chapter 6 to focus on the goal of replicating manual subsystem labellings.

47: Chapter 6: section 6.4: crying out for a precise mathematical explanation of method I have expanded section 6.4 with more detail, including equations 6.1, 6.11, 6.12 and 6.15, and a new diagram, figure 6.4, as well as extra written explanation in section 6.4.1 and section 6.4.2.

48: Chapter 6: Find a way to evaluate cluster quality properly I have added a new section, 6.5, which validates the effectiveness of the Hierarchical Block Matrix / Local Network Learning approach, by comparison with manual subsystem labellings. It uses random shuffling as a baseline, as well as comparing with k-means and walktrap clustering algorithms.

These results are quite promising—compared to the shuffled versions, it get $p < 0.05$ for 5 out of the 8 groups, and it outperforms the baselines of k-means (with a selection of different values of k) and walktrap.