Improving Methods for Analysing Metabolic Pathways in Genome-Scale Metabolic Models Using Mixed Integer Linear Programming

Kayle J. Boessen

Maastricht Science Programme, Faculty of Science and Engineering, Maastricht University

BTR3000 – Bachelor Thesis Research

Research Supervisor: Dr Marian Breuer, Maastricht Centre for Systems Biology

21 May 2024

Wordcount: 6984

Abstract

Constructing biologically accurate *in silico* models of the metabolic functioning of cells in complex organisms like humans is not an easy task. There have been advances in the field of metabolic modelling though. The expression of regulatory genes can be experimentally determined from tissue samples and can then be used to construct genome-scale metabolic models. These models can be analysed using a variety of techniques. This research tries to improve ongoing work in method development to extract task-specific models using mixed-integer linear programming by testing four variations on the current method.

The current method of K Approximation works well for some tasks but not for others, so improvement is needed. While testing different values for ε yielded useful results, improving the LP guess was not successful, signifying that the original method was better. A flux weighted MILP approach showed interesting first results, but needs further modification and testing. A short full model run with the general method shows that roughly 8% of the tasks already pass the test for a correctly functioning model. Concluding that major improvements to the methodology are yet to be attained, but some standards have now been extensively tested and set in place for further research.

Table of Contents

1 - Introduction 4
1.1 - Constraint-Based Analysis 5
1.2 - Human and Tissue-Specific Models 6
1.3 - Metabolic Tasks 7
1.4 - Aim 7
2 - Methods 8
2.1 - Tools and Materials 8
2.2 - General Method 8
2.2.1 - Pre-Processing 8
2.2.2 - FastCC Filtering 9
2.2.3 - K Approximation 9
2.2.4 - iMAT Pruning 13
2.2.5 - Escher-FBA Mapping 13
2.3 - Variations 13
2.3.1 - Epsilon Value Testing 14
2.3.2 - Improving Initial LP Guess 14
2.3.3 - Flux Weighted K Approximation 14
2.3.4 - Full Model Testing 14
3 - Results 16
3.1 - Epsilon Value Tests 16
3.2 - LP Guesses 16
3.3 - Flux Weighted K Approximation 17
3.4 - Full Model Test 18
4 - Discussion 20
4.1 - Limitations 21
4.2 - Future Research and Implications 22
4.3 - Conclusion 23
5 - Critical Reflection 24
5.1 - Acknowledgements 25
6 - References 26
7 - Appendix 29
Appendix A - Figures 29
Appendix B - Tables 32
Appendix C - Files 34

1 - Introduction

Biochemical reactions are an integral part of the functioning of a cell, especially metabolic reactions. Almost all cellular processes rely on the cell's metabolism to supply chemical precursors. Metabolites are constantly flowing into the cell, where they are modified to suit the cell's needs, and then used further or secreted. The metabolic system is a large interconnected network of reactions and pathways. A complex system like this is suitable for mathematical and computational modelling. Computational models allow researchers to perform analysis of the metabolic system *in silic*o[1, 2].

Metabolic systems of small organisms like bacteria and yeast are less difficult to construct compared to human metabolic systems. A core metabolic model of the bacterium Escherichia coli (E. coli) can therefore be used for educational purposes[3]. A visualisation of such a model can be found in Figure 1.

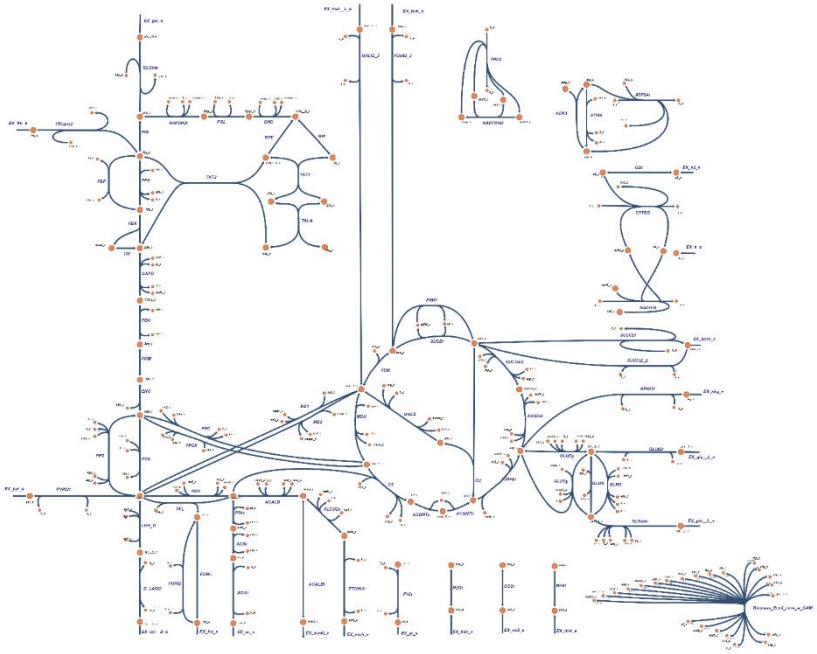

Figure 1. *Metabolic Model of E. Coli.* A metabolic model of the bacterium E. coli[3]. The model includes exchange reactions (labelled EX_) which represent the flow of metabolites into and out of the cell and a simulated biomass function which represents the consumption of metabolites needed for growth. This diagram was created using the visualisation tool Escher-FBA[4]. A higher resolution version can be found in Appendix A.

Metabolic models have evolved in the past decades to not just include metabolic reactions, but also the underlying genetic encoding. This makes it possible to investigate the relationship

between genotype and phenotype[1, 2]. A genome-scale model (GEM) is a computational description of mass-balanced metabolic reactions based on stoichiometry. It also includes gene-protein-reaction associations, based on experimentally obtained information. Greater availability of biological data and technological advances over the recent decades have increased the efficacy of GEMs. Various applications of GEMs have risen over time; suggesting metabolic drug targets in pathogens, predicting new or additional functions of enzymes, and providing - albeit limited - insights into metabolic dysfunction in relation to human diseases[5].

1.1 - Constraint-Based Analysis

An instrumental step in reconstructing and analysing GEMs is to introduce constraints. The metabolic network is first converted into a stoichiometric matrix, which is the central component of a constraint-based model. In such a matrix, the metabolites (m) are represented by the columns and the reactions (r) by the rows, giving the matrix the dimensions $m \times r$. The flow of metabolites, hereafter referred to as the flux, through the network can then be analysed using flux-balance analysis (FBA). FBA is based on the idea that the cell has an objective function towards which it is working. An example of this objective function in the previously mentioned model of E. coli can be the biomass function (see Figure 2). One of the constraints that is added to the model is a steady-state assumption. At this steady state there is mass-balance within the network. This means that net result of the incoming and outgoing metabolites must be zero. Mathematically, this constraint is represented as (1), where S is the stoichiometric matrix and v is the vector containing the fluxes of every reaction in the model (thus having the dimensions $r \times 1$).

$$S \cdot v = 0 \tag{1}$$

The reactions in a model each have a range of potential flux values. These values can also be constrained by setting upper and lower bounds. This will define the solution space of the model. Further constraints can fix cellular uptake and secretion by setting bounds on the flux of exchange reactions, and simulate genetic knockouts by setting the associated reaction bounds to zero. All these constraints and mathematical representations of reactions define a system of linear equations. These equations are then solved using linear programming (LP). When there is more than one solution possible to get to the chosen objective function, alternate methods can be applied to find them. Examples are flux variability analysis (FVA) and algorithms based on mixed-integer linear programming (MILP). Even though FBA can

produce accurate solutions, it also has its limitations. For example, it cannot predict metabolite concentrations and can only determine fluxes in a steady state. Furthermore, FBA in itself does not take regulatory effects like gene expression into account[6-8].

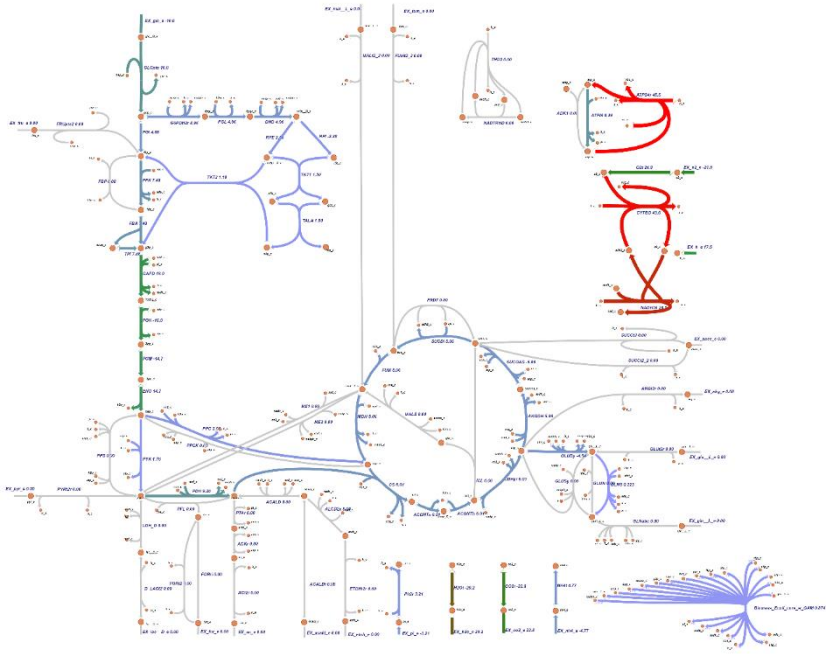

Figure 2. *Flux Balance Analysis on E. Coli.* This diagram shows the result of FBA on a model of the bacterium E. coli. The objective function of this analysis is the simulated biomass function in the bottom right corner. The coloured lines indicate flux, with thicker lines indicating higher flux. This diagram was created using the FBA function of the visualisation tool Escher-FBA[4]. A higher resolution version can be found in Appendix A.

1.2 - Human and Tissue-Specific Models

Creating a GEM for human metabolism is understandably not a simple task. Robinson et al. (2020) were able to unify and improve two model lineages; the Recon series, and the Human Metabolic Reaction series. These model lineages were developed in parallel and influenced each other heavily. The combinatory model was presented as Human1 in 2020. It was constructed and curated using a Git repository, ensuring insight into past and future modifications, as well as providing a platform for community-driven, open-source development[9]. The Human1 model has been receiving updates ever since.

GEMs are constructed to include all cellular metabolic reactions of an organism. However, research shows that not all enzymes are active in every tissue or cell type[10]. To more accurately represent cell metabolism in individual tissues, algorithms have been developed to extract tissue-specific models (also referred to as context-specific models)[11]. The integrative

Metabolic Analysis Tool (iMAT) is one such algorithm. It uses gene expression data and a model to create a map of the metabolic state of that model. This map shows the most likely predicted metabolic fluxes over the reactions in the model[12]. mCADRE (metabolic Context-specificity Assessed by Deterministic Reaction Evaluation) is another tissue-specific model extraction algorithm. This algorithm uses metabolic network topology as well as gene expression to construct models, while also evaluating functionality in the process. mCADRE ranks reactions on their gene expression, the top ranked reactions are seen as the core set. Reactions are removed from the bottom, making sure that the core reactions still function. It has been shown that mCADRE can significantly reduce model construction time[13]. Opdam et al. (2017) used iMAT, mCADRE, and four additional algorithms (MBA, GIMME, INIT, and FastCore) to extract tissue-specific models. They were able to show that these tissue-specific models give more accurate predictions in cell genotype-phenotype relationships than general GEMs[11].

1.3 - Metabolic Tasks

One needs to be careful when interpreting hypotheses on how the metabolic system functions, because there are different methods to create context-specific models. The lack of consensus in curating models led Richelle et al. (2019) to devise an approach that includes the metabolic capabilities of different tissues in models. They created and standardized a list of known metabolic tasks for every cell type[14]. A metabolic task is defined as "a nonzero flux through a reaction or through a pathway leading to the production of a metabolite B from a metabolite A" by Thiele et al. (2013, p. 421). An example of such a task is the aerobic rephosphorylation of ATP from glucose. The curated metabolic task list covers seven major metabolic functions within a cell; energy generation, nucleotide, carbohydrate, amino acid, lipid, vitamin and cofactor, and glycan metabolism. The CellFie framework was then developed to quantify these metabolic functions based directly on transcriptomic data. It identifies the reactions and associated genes from GEMs that are required to perform a metabolic task[16].

1.4 - Aim

The end goal of the overall project is to improve the biological accuracy of the solutions that are calculated using the method currently under development. The aim of this thesis research specifically was to establish definitive values and calculation methods for certain parameters, and to evaluate various possible directions for the future development of the analytical method.

2 - Methods

2.1 - Tools and Materials

For the following methods, the Human1 model version 13.0[9] was used as a base model. A metabolic task list containing 349 metabolic tasks has been curated at the Maastricht Centre for Systems Biology (MaCSBio), this list was also used for this research. A reference list with the descriptions of relevant tasks for this thesis can be found in Appendix B, Table 3. Anonymous gene expression data for this research, consisting of 332 samples experimentally taken from heart muscle tissue, cardiomyocytes, was also supplied by MaCSBio. For all runs in this research, only the expression values from sample 1 were used.

Coding was done in MATLAB version R2023b[17]. For the optimisation the Constraint-Based Reconstruction and Analysis (COBRA) Toolbox version 3.0[18] was used, which includes the Fast Consistency Check (FastCC) algorithm[19]. This algorithm tests the consistency of a stoichiometric model. The Gurobi Solver version 11.0.1[20] was used to solve the MILP problems. To be able to analyse the accuracy of the produced models later on, maps were made to be used with the Escher-FBA[4] visualisation tool. Parts of the code used for the following methods were derived from the CellFie framework and other work by Richelle et al. (2019)[14].

Representative runs were performed on Linux (Ubuntu desktop) virtual workspaces on the SURF Research Cloud. Simple runs were performed on a single core workspace with 4GB RAM memory and more computationally heavy runs were performed on a 16 core workspace with 250GB RAM memory. Comparative runs between these two workspaces were also done to see how much difference using more cores and higher memory makes.

2.2 - General Method

This section explains the general method that was used to generate task-specific solutions from the Human1 model, the gene expression data, and the metabolic task list. The tested variations will be explained in further detail in section 2.3.

2.2.1 - Pre-Processing

The Human1 model, the gene expression data, and the metabolic task list are converted into four separate MAT-files containing the following: the base model, the expression values for each reaction in each sample, the significance of these expression values, and the data for each task. First, the expression values are multiplied by their significance. The expression

data is then adjusted based on whether or not a reaction has associated genes, this is done to help remove orphan reactions. These are reactions that are known to occur, but don't have an expression value because it is unknown what gene product catalyses the reaction[21]. To this end, the orphan reactions are assigned an expression value of -1.

An empty model is created from the base model using the COBRA toolbox. This model is then reduced to a task specific model for each task that was included in the run. The expression values are separated based on the samples that were included in the run, and then further adjusted to match the task specific model. The model is checked using an adapted version of the *checkMetabolicTasks* function in the COBRA toolbox to verify that all metabolites exist and are defined only once within the model. If the associated LP problem is still solvable when the temporary sink reactions that are associated with every in- and output listed in the task are the only exchange reactions that are allowed to carry flux, the model passes[14]. The algorithm only continues with the FastCC filtering and the rest of the pipeline if this check is passed successfully.

2.2.2 - FastCC Filtering

The FastCC algorithm then creates a flux-consistent network by converting all reversible reactions into two irreversible reactions and then optimising for fluxes away from zero. This is all done to reduce the size of the matrix of the task specific model that is used in the K Approximation to a flux consistent version. It also removes or blocks all reactions with a flux smaller than a threshold, which is set to 0.000001. After FastCC filtering, the expression values of the orphan reactions are adjusted to be equal to the median of all expression values greater than zero.

2.2.3 - K Approximation

A new variable, k, is introduced for the K Approximation algorithm. For the approximation of this k, the space containing all possible solutions of the MILP problem that satisfy the constraints is referred to as the feasible space. Likewise, the space with possible values for k that do not satisfy the constraints is the infeasible space (see Figure 3a). To begin defining these solution spaces, the lower bound of the infeasible space is set to the maximum gene expression in the task-specific model, the infeasible space has no finite upper bound. The lower bound of the feasible space is 0 and the upper bound is determined by the initial LP guess, which will be explained in the next paragraph. The K Approximation algorithm

attempts to narrow down the untested space between the feasible and the infeasible solution space by iterating over different values for k in steps equal to 10% of the remaining untested space (see Figure 3b).

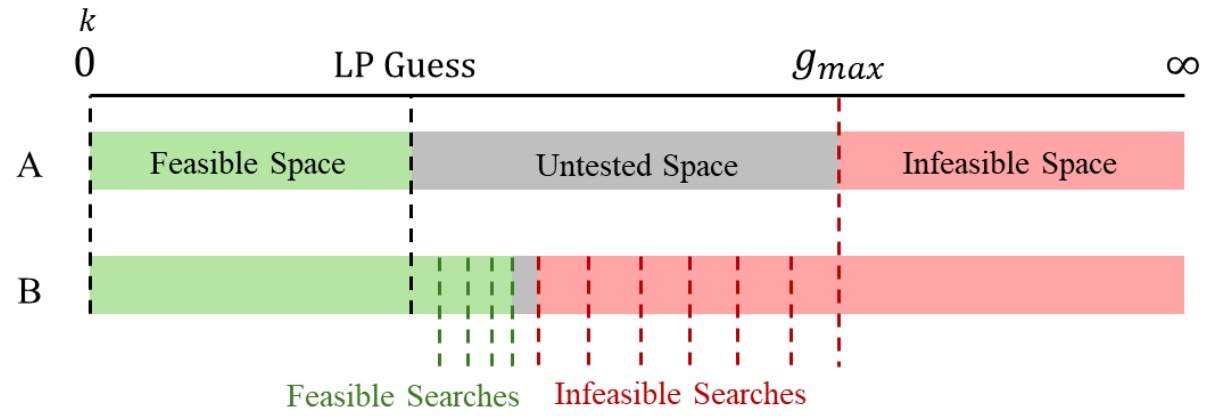

Figure 3. *K Approximation Scheme.* (**A**) The feasible (green), untested (grey), and infeasible (red) space visualised schematically by k value from 0 to ∞. (**B**) The K Approximation algorithm iterates over k values in the untested space to reduce the gap between the feasible and infeasible space.

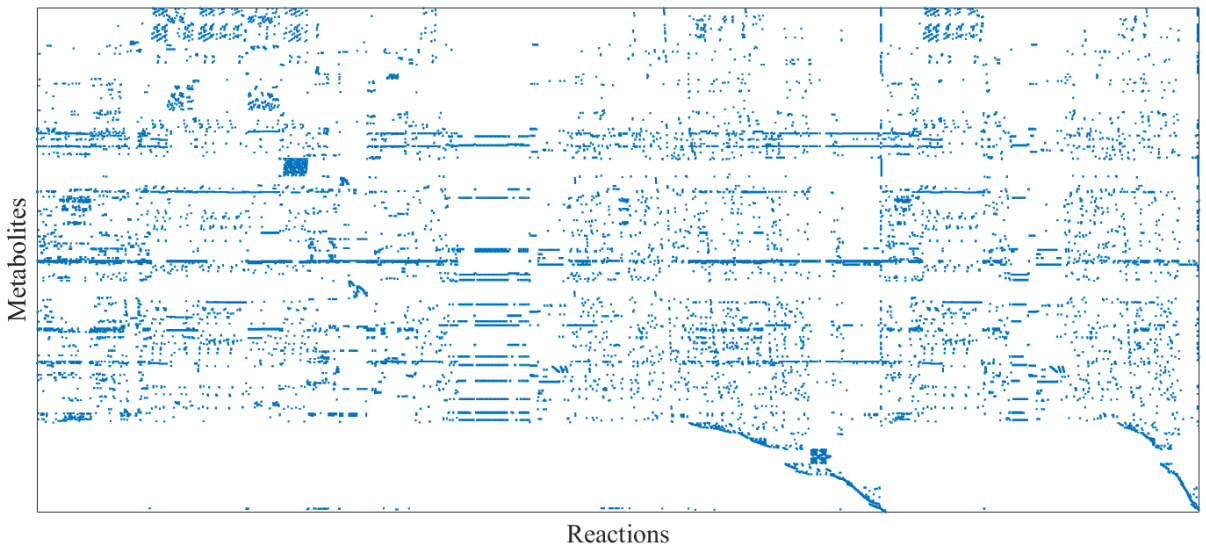

Figure 4. *Stoichiometric Matrix of Task 1.* This figure shows the stoichiometric matrix for task 1. This matrix is part of the model that is fed to the LP solver.

The algorithm then calculates a k value using an LP solver. The model is optimised by the *optimizeCbModel* function of the COBRA toolbox. This function solves the LP problem with the objective (2), where v_i is the flux of reaction i and g_i is the corresponding gene expression value.

$$\min\left(\sum_i \left(v_i \frac{1}{g_i}\right)\right) \tag{2}$$

The expression values from the solution are then used to calculate the first k value with (3). In this equation reaction activity is represented by the binary variable Y_i, which is 1 if the reaction is active and 0 if the reaction is inactive. This k value is thus an average of the gene expression of active reactions in the solution.

$$k = \frac{\sum_i (g_i Y_i)}{\sum_i Y_i} \tag{3}$$

The flux threshold for activity in this calculation is set to 0.001, which means that reactions with an absolute flux greater than 0.001 are considered active. The resulting k value is referred to as the initial LP guess. This value now sets the upper bound of the feasible space (see Figure 3).

For the MILP problem formulation, extra variables and constraints are added as columns and rows to the stoichiometric matrix, as can be seen by comparing Figures 4 and 5. The variable Y_i has an opposite counterpart: Y_i'. This is the inactivity variable, which is 1 if a reaction is inactive, and 0 if the reaction is active. Because the model only contains irreversible reactions after the modifications by the FastCC filtering, a constraint is added that ensures that the two irreversible reactions a formerly reversible reaction is now split into cannot both be active at the same time. The constraints on these variables can be represented as such:

$$Y_i + Y_i' = 1 \tag{4}$$

$$Y_i \leq 1 \tag{5}$$

Three more constraints are added that include the lower bounds ($v_{min,i}$) and upper bounds ($v_{max,i}$) of the flux values. Mathematically, these constraints are represented as (6), (7), and (8). The variable ε in (6) defines a threshold for activity, this is set to 1.000 for all general runs, see also section 2.3.1.

$$v_i + Y_i \times \left(v_{min,i} - \varepsilon\right) > v_{min,i} \tag{6}$$

$$v_i + Y_i \times \left(v_{min,i}\right) > v_{min,i} \tag{7}$$

$$v_i + Y_i \times \left(v_{max,i}\right) < v_{max,i} \tag{8}$$

A further constraint adds the k value to the model, this is mathematically represented in (9).

$$\sum_i \left((g_i - k) \times Y_i\right) \geq 0 \tag{9}$$

The sum of the gene expressions of all active reactions, minus k, is now constrained to be positive. This constrained is only a single row at the bottom of the expanded matrix, where other constraints can be considered as 'blocks' of the matrix, as can be seen in Figure 4.

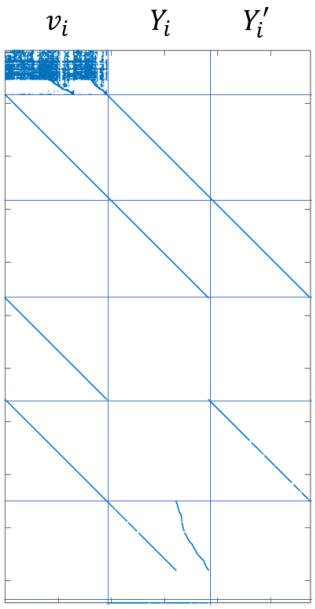

Figure 4. *Expanded Stoichiometric Matrix of Task 1.* This figure shows the stoichiometric matrix for task 1, with added constraints. This matrix is part of the model that is fed to the MILP solver.

The MILP problem can be solved for an objective function, like a maximisation of (3), but this can take a long time. The K Approximation algorithm takes a different approach by using the Gurobi solver to iteratively search for solutions with k values that satisfy constraint (9). The untested space is searched for solutions downwards from the lower bound of the infeasible space. If it does not find solutions, it proves that the model is infeasible within that space and sets the tested k value as the new lower bound for the infeasible space. In case it unexpectedly does find a feasible solution, that solution is stored. The corresponding k value is then set as the new upper bound for the feasible space, and the untested space is searched further from the infeasible side. If the solver reaches the previously set time limit (1200s), it continues to the next part of the approximation.

Next, the untested space is searched iteratively from the upper bound of the feasible space upwards. When a new solution is found, the tested k value is set as the new upper bound, and the average gene expression of active reactions is calculated. The solution is stored as an intermediate solution. This search continues until the solver reaches the time limit (1800s) twice or the difference between the upper and lower bound of the untested space is smaller than the tolerance (0.001). If no solutions are found, which indicates that the LP guess was too high, the upper bound of the feasible space is reduced and the search continues.

Finally, the untested space is searched from both sides again with the default settings of the Gurobi solver to try to narrow the gap between the upper and lower bounds even more. If the difference between the bounds after these rounds is smaller than the tolerance, the run is categorized as converged.

2.2.4 - iMAT Pruning

The model from the last solution of the K Approximation is processed further using iMAT and then stored. The iMAT algorithm extracts a new model by finding the optimal trade-off between removing low-expression reactions and including high-expression reactions. The original iMAT algorithm is adapted for this research to ensure the removal of any remaining orphan reactions.

2.2.5 - Escher-FBA Models

Lastly, all model solutions are prepared for analysis in the Escher-FBA visualisation tool. This is done by running Python code with the COBRA toolbox to convert the solutions from MAT-files to JSON-files. These JSON-files can then be uploaded to the visualisation tool and mapped out to show the reactions, metabolites, and fluxes. This allows for comparison to literature as well as comparisons between different approaches and runs. These Escher-FBA maps were not used directly in this research, but were provided for other project team members.

2.3 - Variations

Several parameters were changed and tested to optimize the general method. The variable ε was given a couple of test values, and the calculation of the initial LP guess was changed. A different MILP formulation was tested, as well as the general method with the full model instead of a task-specific model. The following subsections describe these variations in more detail.

2.3.1 - Epsilon Value Testing

At the beginning of this research, the developing code, which had been cleaned up by previous bachelor thesis students, was returning improper solutions. These solutions were compared to proper solutions that were found by the students previously and the code itself was compared to older versions. The problem seemed to be a change in the value of variable ε. This value of ε is added or subtracted from the lower bound of the flux in the MILP problem constraint (6), so that it acts as a threshold for reaction activity. To find the correct value for future use, four different values of ε were tested: 0.001, 0.010, 0.100, and 1.000. These tests were performed on task 1.

2.3.2 - Improving Initial LP Guess

In order to get a higher k value for the initial LP guess, changes were made to the handling of orphan reactions. Instead of setting the median of all expression values as values for the orphan reactions before the LP solver, the solver was fed the expression values of -1 for the orphan reactions. The expression values were still replaced by the median before going into the MILP solving.

2.3.3 - Flux Weighted K Approximation

To try to improve the biological accuracy of the solutions found by the MILP solver, an alternative MILP formulation was devised. This formulation changes constraint (10) to (11), so that the flux value is included as well.

$$\sum_i (v_i \times g_i - k \times Y_i) \geq 0 \qquad (11)$$

The flux and gene expression of each reaction are multiplied together and the k value is subtracted from the product of this multiplication if the reaction is active. The sum of this subtraction over all reactions is constrained to be positive. This constraint incentivises high gene expression as well as high flux.

The flux weighted K Approximation was tested on tasks 1 and 77. Figure 6 shows how this variation branches off from the general method's pipeline.

2.3.4 - Full Model Testing

Another approach that was tested in this research was a full model run. For this run, the full model was fed into FastCC Filtering and the K Approximation algorithm, instead of task-specific models. The time limits for the infeasible and feasible runs were adjusted to 36,000s

and 54,000s respectively, to take into account the new size of the MILP problem. An overview of the pipeline to test this variation can be found in Figure 6.

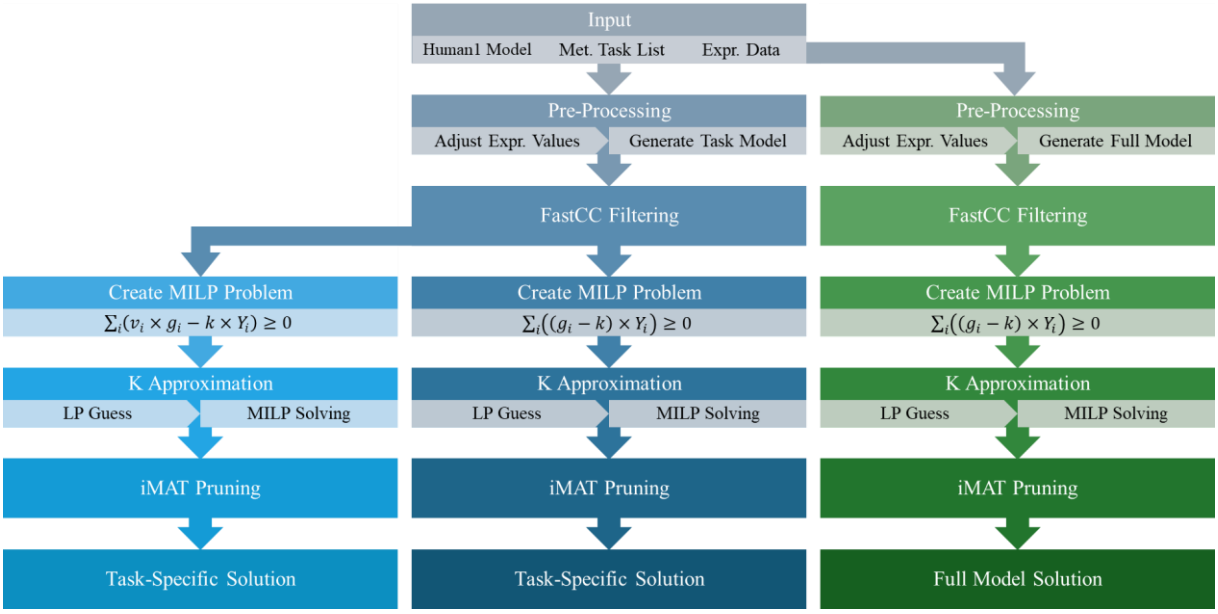

Figure 6. *Method Pipelines.* Pipelines for the three main methods, the general method (dark blue), the full model variation (green), and the flux weighted MILP constraint variation (light blue). This figure shows how the solutions are produced from the input data.

3 - Results

3.1 - Epsilon Value Tests

The four runs were all solved and the results are summarized in Table 1. The run with ε set to 1.000 solved the fastest, in 21364.40 seconds, a little over 5 hours and 56 minutes, with a k value of 4.704. The run with value 0.001 finished faster than the runs with values 0.010 and 0.100, but it also had a larger untested space left and a lower final k value.

ε	No. of Searches		Solve Time (s)	No. of reactions	k	Gap	g_{avg}
	Feasible	Infeasible					
0.001	12	23	28646.11	75	4.932	0.3499	6.237
0.010	18	24	44267.61	74	4.970	0.1903	6.307
0.100	21	25	34799.59	72	4.967	0.1084	6.351
1.000	3	38	21364.40	53	4.704	0.0001	6.046

Table 1. *Results of ε Testing.* The results of the K Approximation of task 1 with different values for ε. The solve time is the duration of the K Approximation. The gap represents the difference between the upper bound of the feasible space and the lower bound of the infeasible space. g_{avg} is the calculated average gene expression of active reactions in the solution model.

The only value of ε for which the K Approximation converged is 1.000, the difference between the feasible and infeasible space for that run is lower than the k tolerance of 0.001. The other values returned higher k values, and they also included more reactions in their final models. All final models are available as MAT-files in Appendix C (Model 1, 2, 3, and 4).

3.2 - LP Guesses

The alternative method to calculate initial LP guesses runs equally fast compared to the general method. The resulting k values are higher than the general method for the tested tasks, as can be seen in Table 2. Among the tested tasks, only tasks 1 and 50 converged.

Task	k Initial LP Guess		Final k MILP Solution
	General	Alternative	
1	3.7316	5.5902	4.704106
5	3.4105	5.2919	5.012095
50	3.8329	6.5506	4.148291
77	3.2110	6.0693	4.470927
84	2.8337	4.5521	4.423454

Table 2. *Results of LP Guesses.* The results of the general and alternative methods for calculating the initial LP guess for tasks 1, 5, 77, and 84. The MILP solution of the general method is also included in the table.

As can also be seen in Table 2, the alternative LP guess for each tested task is higher than the *k* value in the final MILP solution.

3.3 - Flux Weighted K Approximation

The runs with the flux weighted MILP formulation of task 1 and 77 converged very quickly, requiring only one infeasible search to find a solution. Task 1 was solved in 20.67 seconds and task 77 in 1192.51 seconds, both runs were done on the single core workspace. Both task-specific solutions passed the *checkMetabolicTasks* function. FVA was performed on both tasks, the results of which can be seen in Figures 7 and 8.

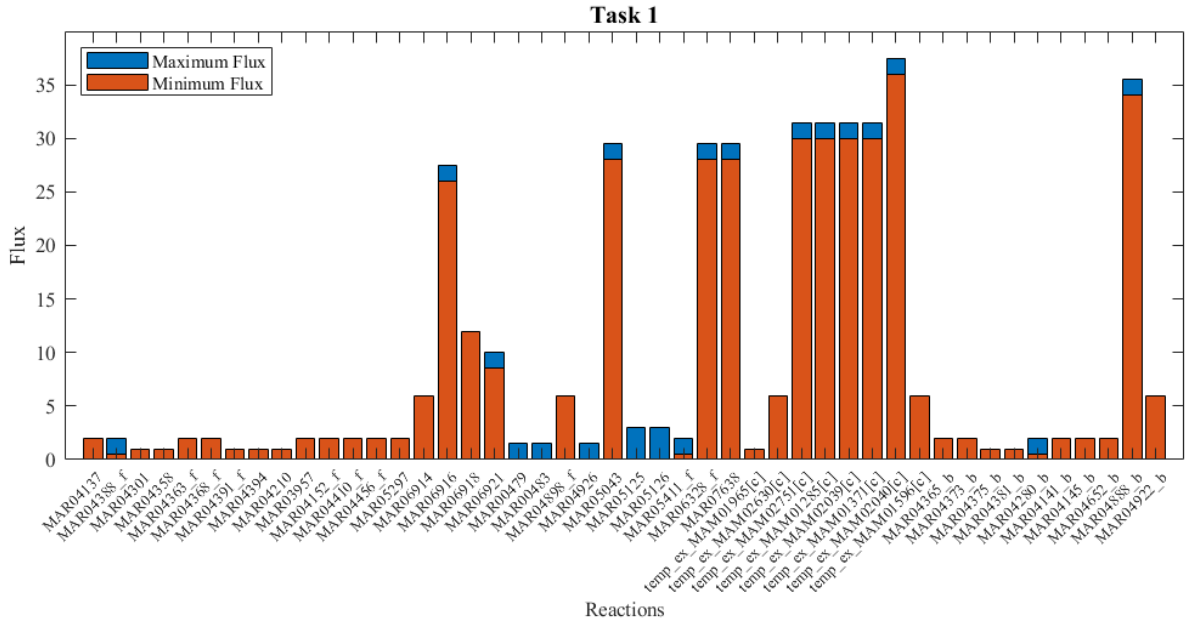

Figure 7. *Flux Variability Analysis Task 1.* This bar plot shows the results of FVA on the task 1 solution of the Flux Weighted K Approximation. The maximum flux is shown in blue and the minimum flux in red. The reactions are described by their Human-GEM reaction ID.

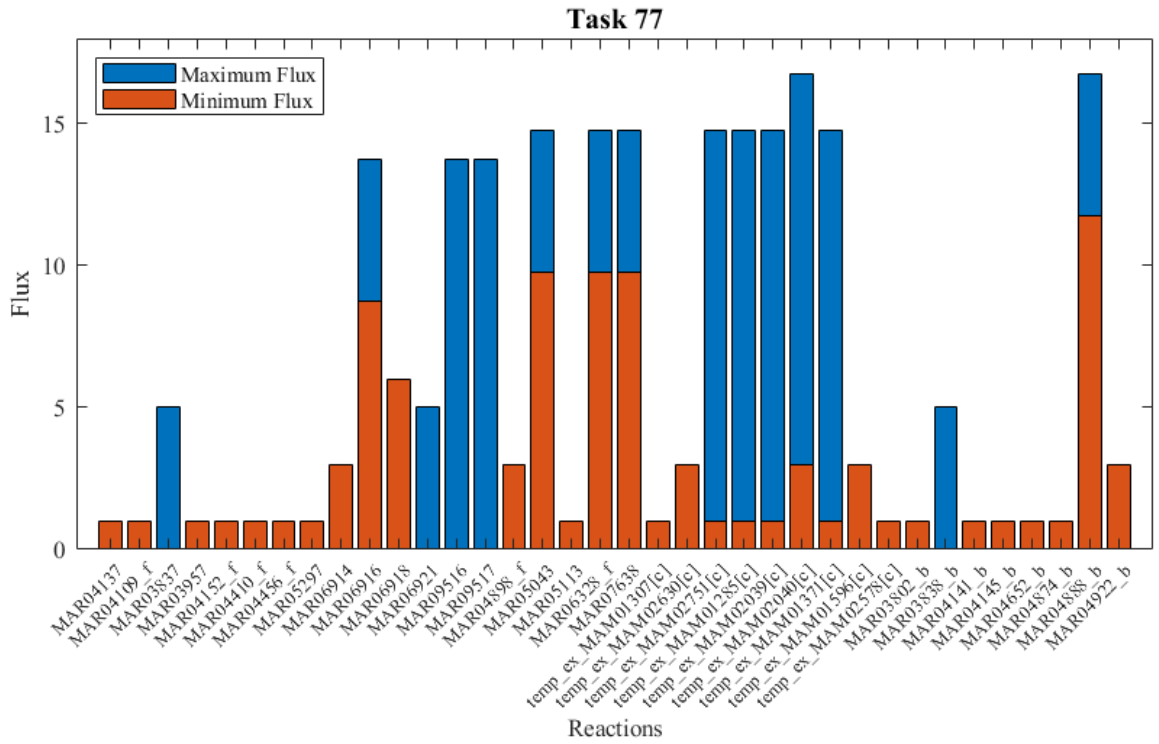

Figure 8. *Flux Variability Analysis Task 77.* This bar plot shows the results of FVA on the task 77 solution of the Flux Weighted K Approximation. The maximum flux is shown in blue and the minimum flux in red. The reactions are described by their Human-GEM reaction ID.

The number of reactions in the solution for task 1 was 46, with a *k* value of 3.744 and an average gene expression of 5.023. For task 77, 36 reactions were included, with a *k* value of 3.246 and an average gene expression of 4.812. The final models of the flux weighted K Approximation for both tasks are available as MAT-files in Appendix C (Model 5 and 6).

Extended runs with a higher initial infeasible *k* test value were also done. These runs did not have enough solving time to find any solutions, and were therefore deemed inconclusive.

3.4 - Full Model Test

Of the original 13026 reactions in the Human1 model, 758 were included in the solution of the full model run. The run did not converge, and was terminated due to an error in one of the subfunctions of the algorithm. The last feasible solution was found with a *k* value of 2.049 and an average gene expression of 2.832.

The full model solution was tested for all 349 tasks using the *checkMetabolicTasks* function, and 30 of the 349 tasks passed. A list of the tasks that passed can be found in Table 4 in Appendix B. FVA was performed on the full model solution, the results of which can be seen

in Figure 9. This data is also available as a CSV file (File 3 in Appendix C), and the last feasible solution model is available as well as a MAT file (Model 7 in Appendix C).

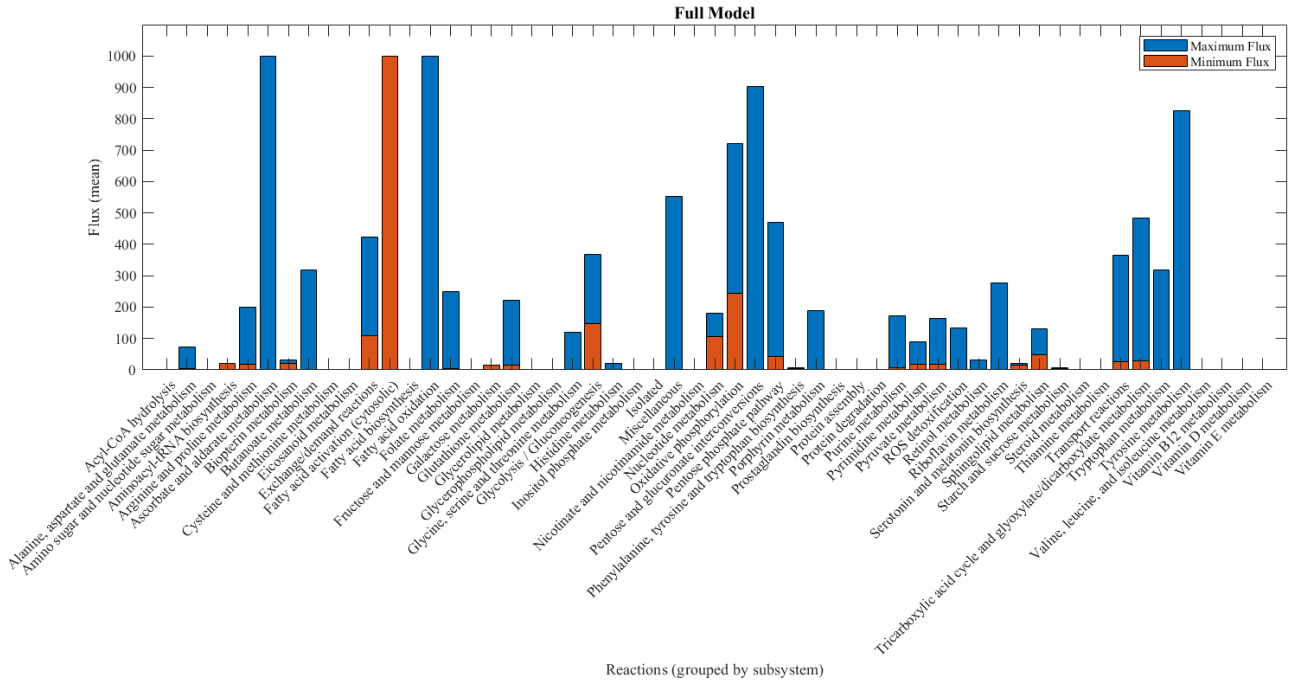

Figure 9. *Flux Variability Analysis Full Model.* This bar plot shows the mean results of FVA on the full model solution of the general method. The maximum flux is shown in blue and the minimum flux in red. The reactions are grouped by subsystem, the means are calculated per subsystem.

4 - Discussion

Running with different epsilon values shows how much the activity threshold matters in refining the final task model. With a flux threshold lower than 1, around twenty more reactions are added to the solution to come up to a higher average expression. When comparing the reactions of the solutions with ε set to 1 and to 0.001, we find that 25 reactions are removed. An example is a cycle that oxidises vitamin C (ascorbate) using iron(III) and replenishes the iron(III) through ferroxidase. Its removal is good because that cycle doesn't make sense for the task biologically. For 28 reactions, the fluxes are changed and three reactions are included in the $\varepsilon = 1$ solutions, that are not in the $\varepsilon = 0.001$ solution (for full comparison see File 1 in Appendix C). Looking at literature, Shlomi et al. (2008), one of the papers that other methods like iMAT and INIT are based on, also uses ε = 1 as an activity threshold[7]. It was decided that a value of 1.000 would be used for ε for every run thereafter.

On first look, the alternative method to calculate LP guesses has promising results. The LP guess for each task is higher than that of the general method. This is because orphan reactions, which have a gene expression value of -1, are not adjusted to the median gene expression of all reactions. Thus, they may not be included in the LP solution as the solver optimizes for a maximum average gene expression. If the alternative LP guess is higher than when using the general method, the untested space is smaller, which could speed up the K Approximation. However, at least for the tested tasks, the LP guess in the alternative method is higher than the final solution value for k. In these cases, the feasible search starts too high. This then results in the K Approximation taking longer because the upper bound of the feasible space needs to be reduced to find any solution. Other approaches to reduce solving time may also be better because the LP guess then at least uses the same expression values as the MILP problem.

Though the runs with the flux weighted MILP formulation converged faster than the general method, these results are skewed to an extent. The first k value tested as the lower bound of the infeasible space instantly produced a feasible solution. This means that the infeasible space was not searched thoroughly enough to be able to definitively say that this is the optimal solution. The solutions include fewer reactions, and a lower average gene expression. When comparing the results of the FVA (Figure 7 and 8) to those of the general method (Figure 10 and 11 in Appendix A), one can see that the overall variability of the reactions that

are also present in the general solution is higher. This means that there are more possible solutions that satisfy the constraints.

Running the general method on the full model yielded very basic results, as the K Approximation did not have enough time to converge, partially because the run was terminated due to an error in the code (see also paragraph five of the Limitations). The intermediate solutions were still analysed, and revealed that at least for 30 tasks the algorithm had found a passing solution. The FVA of the solution shows a high flux variability in most active reaction subsystems. It also shows that certain reaction subsystems, like vitamin metabolism, are not included in the solution, as there are no flux values. A higher time limit for the infeasible runs could reduce the untested space further, but the last feasible run already took 9645 seconds. In that last run, almost 16.5 thousand nodes were explored, with one set of 300 nodes taking an hour to complete. It is therefore expected that any more feasible iterations would take a very long solving time to find a solution.

4.1 - Limitations

The methods tested in this research suggest that the enzymes that catalyse the reactions are only driven by the expression of their associated genes. However, the functioning of enzymes can also be described by other variables, like their turnover rate. This rate defines the maximum chemical conversion of the reaction[22]. When the turnover rate of a reactions is high, it may very well have a low gene expression, since that low expression might turn over enough for the cell's needs. The methods don't include this because of the limited number of resources available for turnover rates. An approach to include these turnover rates in this research was planned, but postponed due to the unavailability of the data in the timespan of the research.

Another planned approach that was not implemented during this research involved using methods similar to the INIT algorithm. This research was performed by another bachelor thesis student to split the workload. Such limitations are inevitable in the relatively short period of time that was given for this research.

Once a protein or enzyme is translated from RNA, driven by the gene expression, it can still be modified. Post-translational modifications include the addition of chemical and complex groups on the protein. These kinds of modifications have a wide range of implications for the

structure and functioning of the protein, and they happen continuously on almost all proteins.[23]

The variations on the general method were mainly tested using task 1, as it is a task that easily converges, and that does this in a relatively short amount of time, thus making it ideal to test new approaches. However, it has been noted that other tasks take a lot longer to converge, even with the general method, and some don't converge at all. This can be due to several factors, among which is the specificity of the gene expression data. As all cell types differ in their functions, they also differ in what metabolic reactions are expressed. Furthermore, as only the gene expression data of sample 1 was used, every solution is only valid for the metabolic functioning of a specific tissue of one individual. There is a variety even within samples from the same tissue but from different humans.

A final limiting factor within this research was the time that was available to run the code on the virtual workspaces. With 16 cores running simultaneously for a long time to solve the full model runs, the budget quickly ran out. Therefore it was not possible to rerun the code with greater time limits to try to find more feasible solutions. Using other workspaces to run the code would heavily influence the time results by introducing other variables. For reference, K Approximation for task 1 took 21364.40 seconds on the single core workspace, and 1709.09 seconds on the 16 core workspace.

4.2 - Future Research and Implications

In future research, inclusion of gene expression data from different tissues could be tested. This could result in solutions that give a more generalised view of the metabolic functioning of a human cell. Another option is to include homogenised data from one tissue, for example by incorporating pooled gene expression data from multiple samples that are already available. This should result in broader but still tissue-specific solutions.

The method of using the flux weighted MILP formulation needs more research. A restructuring of the code and testing of more parameters for the K Approximation is necessary to definitively say if the flux weighted formulation works better than the general formulation or not. It is also possible to test different MILP formulations altogether, for example using one over the square of gene expression in the k constraint.

Some post-translational modification data is already available from publicly accessible databases[23], its inclusion could improve the reliability of the solutions. As explained before,

the methods should not just rely on gene expression, but also on other variables that affect the enzymes catalysing the metabolic reactions. If other proteomic data, like enzyme turnover, becomes more widely available

A new MILP algorithm that works more along the lines of the INIT algorithm is under development by others in the research team. This new algorithm has not been tested thoroughly yet, but the preliminary results seem promising. Future research should try to improve on this, as it may be a suitable replacement for the current K Approximation.

On the broad spectrum of metabolic modelling and task analysis, the research in this thesis does not necessarily make a great impact. However, some standards have now been extensively tested and set in place for further research at MaCSBio and beyond.

4.3 - Conclusion

It is now possible to definitively say which value for ε and which method to calculate the initial LP guess should be used. The flux weighted alternative formulation for K Approximation needs more development to be conclusive and the full model test has promising first results. In conclusion, the analytical methodology has ever so slightly improved, but needs much further research.

5 - Critical Reflection

Even though I had a head start, working with the code and joining the project group meetings before the actual start of my BTR, I felt a bit out of my depth at the beginning. This could have been prevented by doing even more background research beforehand. Also, it had been some time since I had done any programming in MATLAB, so to me it felt like a slow start. In terms of time planning, I think I made a lot of improvements in my usual habits by setting clear deadlines for myself. Nevertheless, I feel like I could have contributed more to the project if I had not postponed certain tasks until those I got closer to those self-imposed deadlines.

While testing different parameters and strategies, my initial intention of logging all files and data neatly did not turn out exactly as I hoped. This led to having to look back and extract what I needed for the written thesis from a large amount of semi-organised files, which took a lot of time. By working on two different research clouds and my personal laptop at the same time, I often got confused and at one point permanently deleted a folder of files I thought I had copied to a hard drive, only to find out I had to redo half a day's work. This experience led me to change the way I backed-up all files to GitHub and I got a lot more structured after that.

I also think that I could have asked for help sooner whenever I didn't understand why the results differed from expectations, usually I waited until the weekly meeting to present my findings and ask questions. I think that a more day to day routine instead of a week to week one would have improved this.

Finally, I think that I should have spent less time on cleaning up the code in the beginning. This was time that I could have spent on implementing new approaches. Some cleanup was necessary to restore the code to function as it did before. After that, I could have left the rest of the cleanup for after writing my thesis so that I would have more time to tweak the code and get some more results. On the other hand, cleaning up and working with the code made me more familiar with the custom functions and toolboxes that were used.

All in all I think I made a lot of personal progress and at the same time I learned a lot about the area of systems biology and metabolic modelling in particular.

5.1 - Acknowledgements

First of all, I would like to thank my research supervisor, Dr Marian Breuer, for his teachings, help, and advice, as well as for allowing me to join his team early to get to know the project and background better. My gratitude goes out to Jelle Bonthuis and Justin Cornélis, who both contributed a lot to the code and helped me to understand the work that had been done on the methods previously. I would also like to thank the rest of our project team: Tom, Sorismonde, and Jana for their supportive work on the project. I'm very thankful to my boyfriend Nohr for his moral support and for helping me revise this thesis. Finally, I would like to thank the staff and other interns at MaCSBio for the pleasant work environment and the great social gatherings.

6 - References

1. Lewis, N. E., Nagarajan, H., & Palsson, B. O. (2012). Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods. *Nature Reviews Microbiology*, *10*(4), 291-305. https://doi.org/10.1038/nrmicro2737
2. O'Brien, Edward J., Monk, Jonathan M., & Palsson, Bernhard O. (2015). Using Genome-scale Models to Predict Biological Capabilities. *Cell*, *161*(5), 971-987. https://doi.org/https://doi.org/10.1016/j.cell.2015.05.019
3. Orth Jeffrey, D., Fleming, R. M. T., & Palsson Bernhard, Ø. (2010). Reconstruction and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide. *EcoSal Plus*, *4*(1), 10.1128/ecosalplus.1110.1122.1121. https://doi.org/10.1128/ecosalplus.10.2.1
4. King, Z. A., Dräger, A., Ebrahim, A., Sonnenschein, N., Lewis, N. E., & Palsson, B. O. (2015). Escher: A Web Application for Building, Sharing, and Embedding Data-Rich Visualizations of Biological Pathways. *PLOS Computational Biology*, *11*(8), e1004321. https://doi.org/10.1371/journal.pcbi.1004321
5. Gu, C., Kim, G. B., Kim, W. J., Kim, H. U., & Lee, S. Y. (2019). Current status and applications of genome-scale metabolic models. *Genome Biology*, *20*(1), 121. https://doi.org/10.1186/s13059-019-1730-3
6. Bordbar, A., Monk, J. M., King, Z. A., & Palsson, B. O. (2014). Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics*, *15*(2), 107-120. https://doi.org/10.1038/nrg3643
7. Shlomi, T., Cabili, M. N., Herrgård, M. J., Palsson, B., & Ruppin, E. (2008). Network-based prediction of human tissue-specific metabolism. *Nat Biotechnol*, *26*(9), 1003-1010. https://doi.org/10.1038/nbt.1487
8. Orth, J. D., Thiele, I., & Palsson, B. Ø. (2010). What is flux balance analysis? *Nature Biotechnology*, *28*(3), 245-248. https://doi.org/10.1038/nbt.1614
9. Robinson, J. L., Kocabaş, P., Wang, H., Cholley, P.-E., Cook, D., Nilsson, A., Anton, M., Ferreira, R., Domenzain, I., Billa, V., Limeta, A., Hedin, A., Gustafsson, J., Kerkhoven, E. J., Svensson, L. T., Palsson, B. O., Mardinoglu, A., Hansson, L., Uhlén, M., & Nielsen, J. (2020). An atlas of human metabolism. *Science Signaling*, *13*(624), eaaz1482. https://doi.org/doi:10.1126/scisignal.aaz1482

10. Uhlén, M., Fagerberg, L., Hallström, B. M., Lindskog, C., Oksvold, P., Mardinoglu, A., Sivertsson, Å., Kampf, C., Sjöstedt, E., Asplund, A., Olsson, I., Edlund, K., Lundberg, E., Navani, S., Szigyarto, C. A.-K., Odeberg, J., Djureinovic, D., Takanen, J. O., Hober, S., . . . Pontén, F. (2015). Tissue-based map of the human proteome. *Science*, *347*(6220), 1260419. https://doi.org/doi:10.1126/science.1260419
11. Opdam, S., Richelle, A., Kellman, B., Li, S., Zielinski, D. C., & Lewis, N. E. (2017). A Systematic Evaluation of Methods for Tailoring Genome-Scale Metabolic Models. *Cell Syst*, *4*(3), 318-329.e316. https://doi.org/10.1016/j.cels.2017.01.010
12. Zur, H., Ruppin, E., & Shlomi, T. (2010). iMAT: an integrative metabolic analysis tool. *Bioinformatics*, *26*(24), 3140-3142. https://doi.org/10.1093/bioinformatics/btq602
13. Wang, Y., Eddy, J. A., & Price, N. D. (2012). Reconstruction of genome-scale metabolic models for 126 human tissues using mCADRE. *BMC Systems Biology*, *6*(1), 153. https://doi.org/10.1186/1752-0509-6-153
14. Richelle, A., Chiang, A. W. T., Kuo, C.-C., & Lewis, N. E. (2019). Increasing consensus of context-specific metabolic models by integrating data-inferred cell functions. *PLOS Computational Biology*, *15*(4), e1006867. https://doi.org/10.1371/journal.pcbi.1006867
15. Thiele, I., Swainston, N., Fleming, R. M. T., Hoppe, A., Sahoo, S., Aurich, M. K., Haraldsdottir, H., Mo, M. L., Rolfsson, O., Stobbe, M. D., Thorleifsson, S. G., Agren, R., Bölling, C., Bordel, S., Chavali, A. K., Dobson, P., Dunn, W. B., Endler, L., Hala, D., . . . Palsson, B. Ø. (2013). A community-driven global reconstruction of human metabolism. *Nature Biotechnology*, *31*(5), 419-425. https://doi.org/10.1038/nbt.2488
16. Richelle, A., Kellman, B. P., Wenzel, A. T., Chiang, A. W. T., Reagan, T., Gutierrez, J. M., Joshi, C., Li, S., Liu, J. K., Masson, H., Lee, J., Li, Z., Heirendt, L., Trefois, C., Juarez, E. F., Bath, T., Borland, D., Mesirov, J. P., Robasky, K., & Lewis, N. E. (2021). Model-based assessment of mammalian cell metabolic functionalities using omics data. *Cell Reports Methods*, *1*(3), 100040. https://doi.org/https://doi.org/10.1016/j.crmeth.2021.100040
17. Inc, T. M. (2023). *MATLAB*. In (Version 23.2.0.2485118 (R2023b) Update 6) The MathWorks Inc. https://www.mathworks.com
18. Heirendt, L., Arreckx, S., Pfau, T., Mendoza, S. N., Richelle, A., Heinken, A., Haraldsdóttir, H. S., Wachowiak, J., Keating, S. M., Vlasov, V., Magnusdóttir, S., Ng, C. Y., Preciat, G., Žagare, A., Chan, S. H. J., Aurich, M. K., Clancy, C. M., Modamio,

J., Sauls, J. T., . . . Fleming, R. M. T. (2019). Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v.3.0. *Nature Protocols*, *14*(3), 639-702. https://doi.org/10.1038/s41596-018-0098-2

19. Vlassis, N., Pacheco, M. P., & Sauter, T. (2014). Fast Reconstruction of Compact Context-Specific Metabolic Network Models. *PLOS Computational Biology*, *10*(1), e1003424. https://doi.org/10.1371/journal.pcbi.1003424
20. Gurobi Optimization, L. (2023). *Gurobi Optimizer Reference Manual*. In (Version 11.0.1) https://www.gurobi.com
21. Orth, J. D., & Palsson, B. (2010). Systematizing the generation of missing metabolic knowledge. *Biotechnol Bioeng*, *107*(3), 403-412. https://doi.org/10.1002/bit.22844
22. Li, F., Yuan, L., Lu, H., Li, G., Chen, Y., Engqvist, M. K. M., Kerkhoven, E. J., & Nielsen, J. (2022). Deep learning-based kcat prediction enables improved enzyme-constrained model reconstruction. *Nature Catalysis*, *5*(8), 662-672. https://doi.org/10.1038/s41929-022-00798-z
23. Ramazi, S., & Zahiri, J. (2021). Posttranslational modifications in proteins: resources, tools and prediction methods. *Database (Oxford)*, *2021*. https://doi.org/10.1093/database/baab012

7 - Appendix

Appendix A - Figures

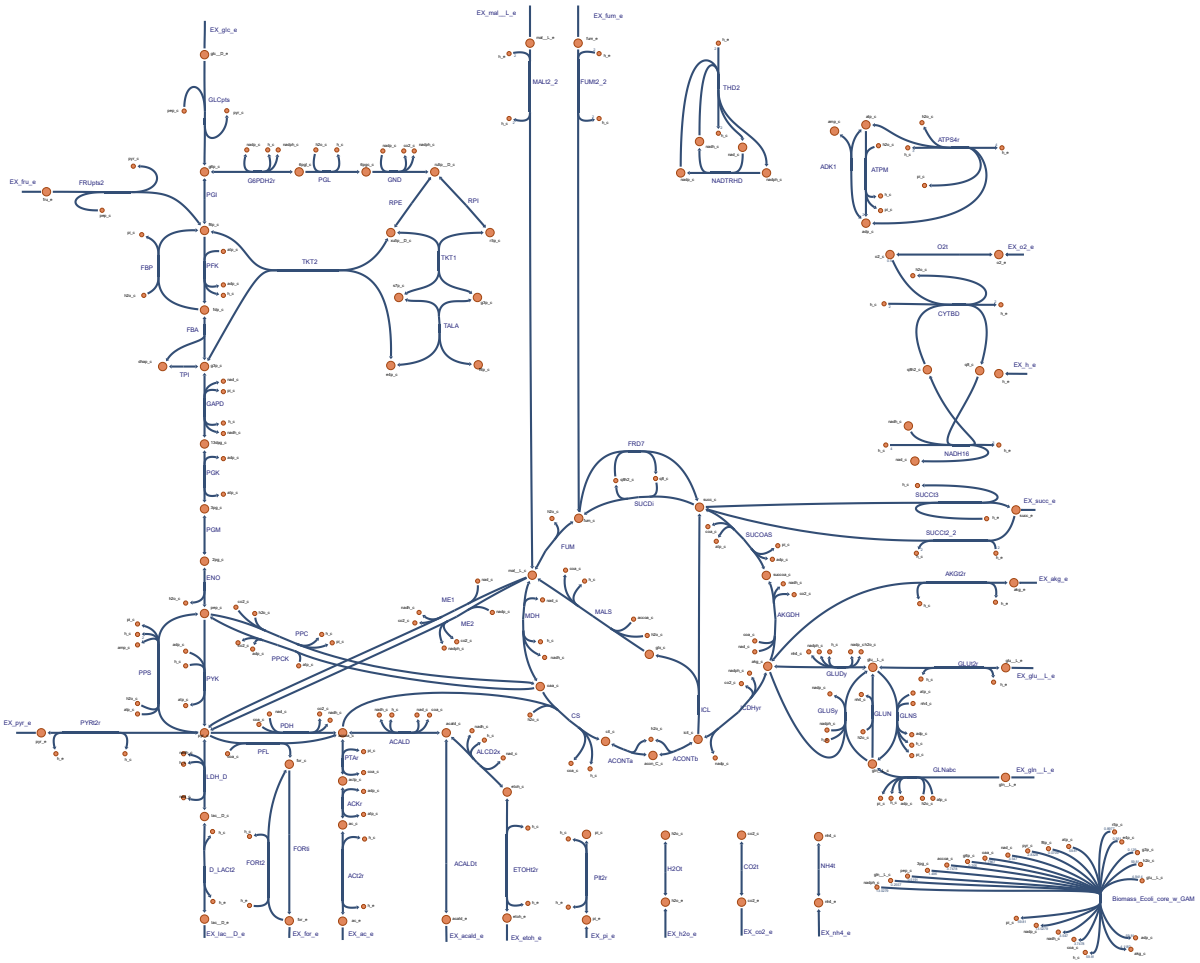

Figure 1 (High Resolution). *Metabolic Model of Escherichia Coli*. A metabolic model of the bacterium Escherichia Coli (E. coli)[3]. The model includes exchange reactions (labelled EX_) which represent the flow of metabolites into and out of the cell and a simulated biomass function which represents the consumption of metabolites needed for growth. This diagram was created using the visualisation tool Escher-FBA[4].

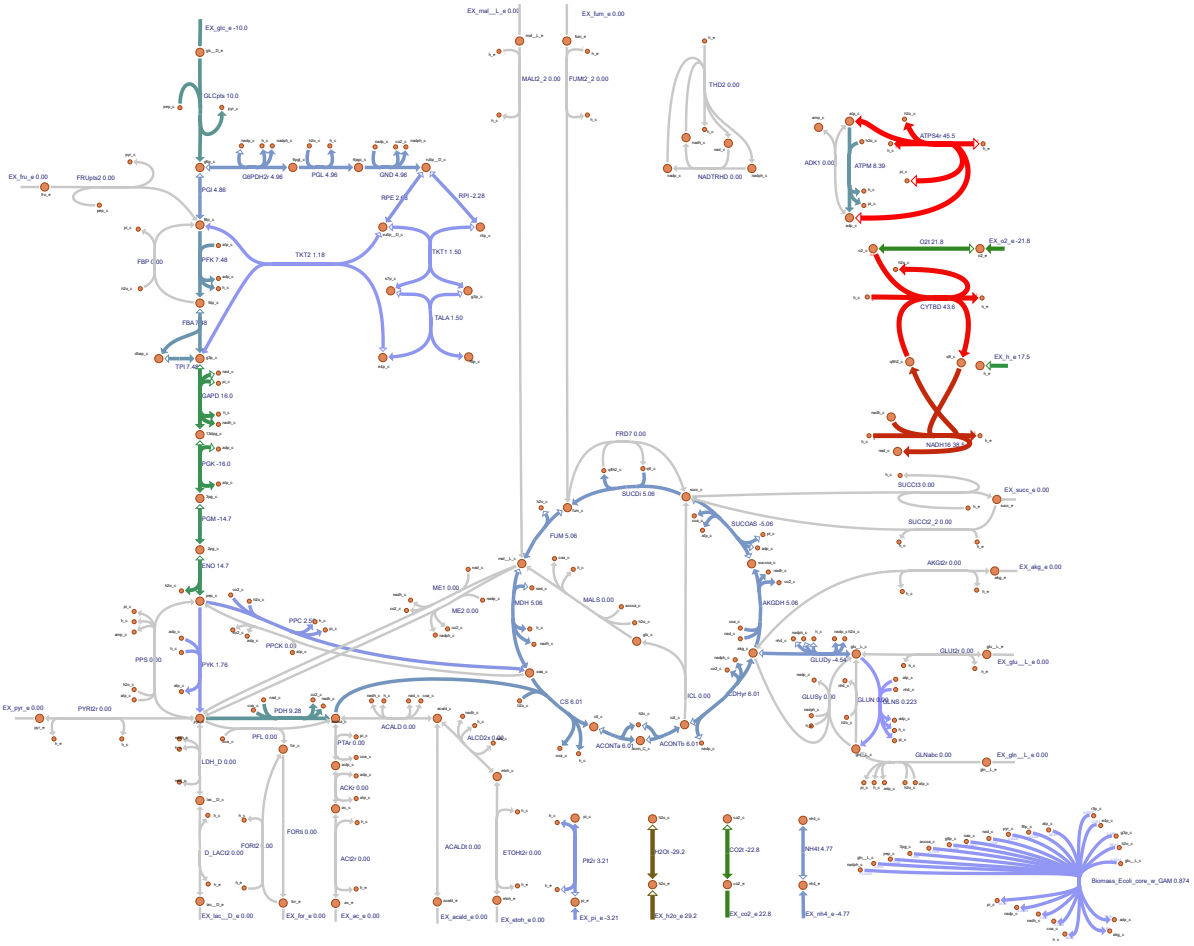

Figure 2 (High Resolution). *Flux Balance Analysis on E. Coli.* This diagram shows the result of FBA on a model of the bacterium E. coli. The objective function of this analysis is the simulated biomass function in the bottom right corner. The coloured lines indicate flux, with thicker lines indicating higher flux. This diagram was created using the FBA function of the visualisation tool Escher-FBA[4].

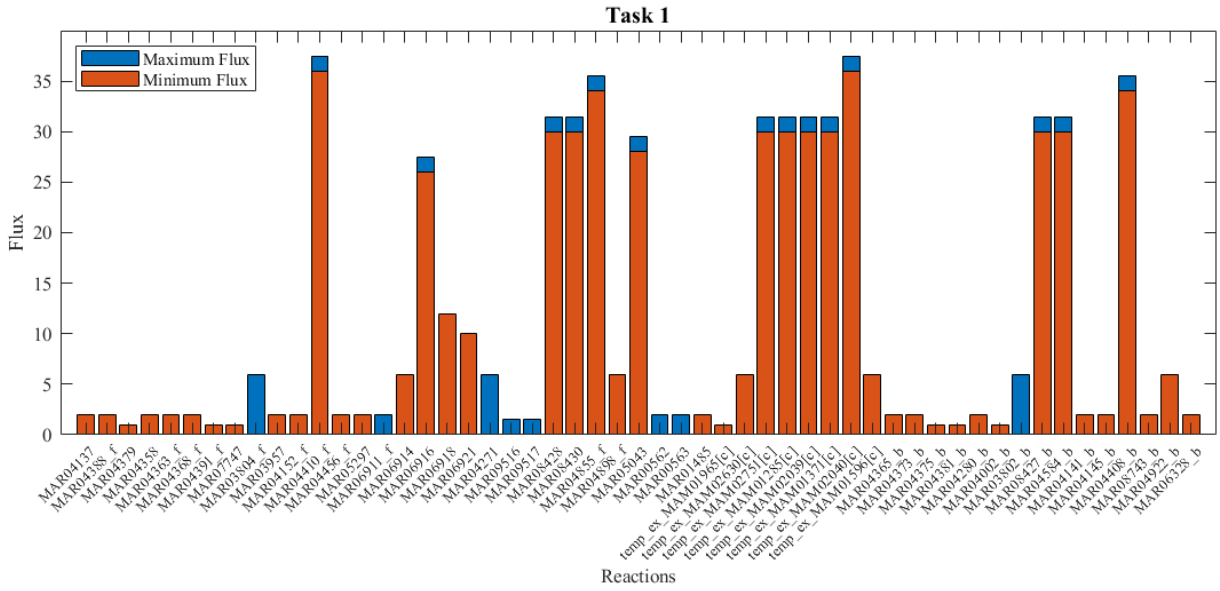

Figure 10. *Flux Variability Analysis Task 1.* This bar plot shows the results of FVA on the task 1 solution of the general K Approximation. The maximum flux is shown in blue and the minimum flux in red. The reactions are described by their Human-GEM reaction ID.

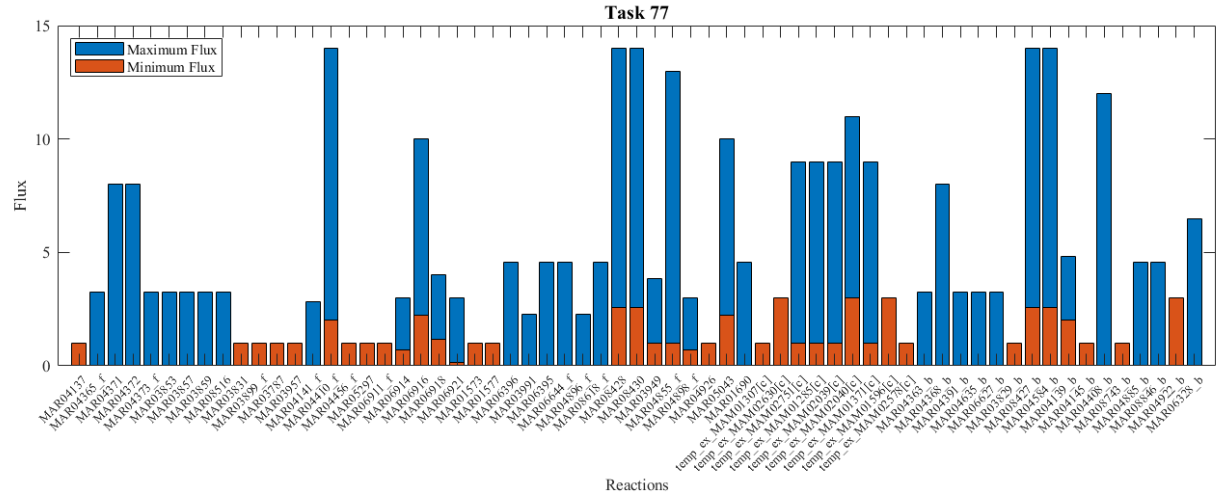

Figure 11. *Flux Variability Analysis Task 77.* This bar plot shows the results of FVA on the task 77 solution of the general K Approximation. The maximum flux is shown in blue and the minimum flux in red. The reactions are described by their Human-GEM reaction ID.

Appendix B - Tables

Task	System	Description
1	Energy Metabolism	Aerobic rephosphorylation of ATP from glucose
5	Energy Metabolism	Aerobic rephosphorylation of UTP
50	Amino Acids Metabolism	Serine de novo synthesis (minimal substrates, minimal excretion)
77	Amino Acids Metabolism	Tyrosine de novo synthesis (minimal substrates, minimal excretion)
84	Amino Acids Metabolism	Histidine degradation

Table 3. *Relevant Metabolic Tasks.* This table contains the systems and descriptions of the tasks that were used to test variations on the general methods.

Task	System	Description
33	Carbohydrates Metabolism	UDP-glucose de novo synthesis (minimal substrates, physiological excretion)
35	Carbohydrates Metabolism	UDP-glucuronate de novo synthesis (minimal substrates, physiological excretion)
73	Amino Acids Metabolism	Tyrosine de novo synthesis (minimal substrates with AA, minimal excretion)
82	Amino Acids Metabolism	Glutamate degradation
83	Amino Acids Metabolism	Glycine degradation
84	Amino Acids Metabolism	Histidine degradation
86	Amino Acids Metabolism	Glutamine degradation
98	Amino Acids Metabolism	Ornithine degradation
102	Amino Acids Metabolism	Creatine de novo synthesis (minimal substrates, physiological excretion + arginine)
113	Vitamin & Cofactor Metabolism	NAD de novo synthesis (minimal substrates, physiological excretion)
115	Vitamin & Cofactor Metabolism	FAD de novo synthesis (minimal substrates, physiological excretion)
247	Amino Acids Metabolism	SAM de novo synthesis (minimal substrates, physiological excretion)
248	Amino Acids Metabolism	GSH de novo synthesis (minimal substrates, physiological excretion)

253	Energy Metabolism	Oxidative decarboxylation
260	Energy Metabolism	Presence of the thioredoxin system through the thioredoxin reductase activity
262	Nucleotide Metabolism	GMP salvage from guanine
271	Carbohydrates Metabolism	Malate to pyruvate conversion
281	Carbohydrates Metabolism	Synthesis of phosphatidylinositol from inositol
286	Amino Acids Metabolism	Synthesis of alanine from glutamine
289	Amino Acids Metabolism	Synthesis of aspartate from glutamine
290	Amino Acids Metabolism	Conversion of aspartate to arginine
292	Amino Acids Metabolism	Conversion of asparate to asparagine
297	Amino Acids Metabolism	Conversion of glutamate to glutamine
298	Amino Acids Metabolism	Conversion of glutamate to proline
299	Amino Acids Metabolism	Conversion of GABA into succinate
302	Amino Acids Metabolism	Conversion of histidine to glutamate
308	Amino Acids Metabolism	Synthesis of spermidine from ornithine
311	Amino Acids Metabolism	Phenylalanine to tyrosine
318	Amino Acids Metabolism	Synthesis of serotonin from tryptophan
326	Lipids Metabolism	Glycerol-3-phosphate synthesis

Table 4. *Passed Metabolic Tasks in Full Model Solution.* This table contains the systems and descriptions of the tasks that passed the *checkMetabolicTasks* function in the full model variation.

Appendix C - Files

File 1. *Reaction and Flux Comparison of ε tests 0.001 and 1.000*

File 2. *Reaction and Flux Comparison of General and Flux Weighted Methods*

File 3. *Mean FVA of Full Model Solution*

Model 1. *Solution Model of Task 1 with ε = 0.001*

Model 2. *Solution Model of Task 1 with ε = 0.010*

Model 3. *Solution Model of Task 1 with ε = 0.100*

Model 4. *Solution Model of Task 1 with ε = 1.000*

Model 5. *Solution Model of Task 1 of the Flux Weighted Method*

Model 6. *Solution Model of Task 77 of the Flux Weighted Method*

Model 7. *Last Intermediate Solution Model of the Full Model Run*